



UNIVERSIDAD CARLOS III DE MADRID

PROYECTO FINAL DE CARRERA

Ingeniería Industrial

**ESTUDIO DE MÁQUINAS DE VECTORES DE SOPORTE PARA
CLASIFICACIÓN DE ELECTROCARDIORAMAS**

Autora: D^a. Olga Sarma Belán

Directora: Dra. D^a. Marta Ruiz Llata

Escuela Politécnica Superior

Departamento de Tecnología Electrónica

Leganés, Julio 2013

Titulo: Estudio de máquinas de vectores de soporte para clasificación de electrocardiogramas.

Autora: D^a. Olga Sarma Belán

Directora: Dra. D^a. Marta Ruiz Llata

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Final de Carrera el día 2 de julio de 2013 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Resumen

En el presente proyecto final de carrera se presenta un algoritmo de clasificación de las señales biomédicas, en concreto las pulsaciones cardíacas de la base de electrocardiogramas (ECGs) MIT Arrhythmia. Las pulsaciones elegidas se componen de siete anómalas y una normal. La dificultad añadida durante la clasificación radica en que dos pulsaciones anómalas se forman a partir de la unión de otras dos pulsaciones, que también entran en la clasificación.

El proyecto empieza con el repaso de los fundamentos teóricos del aprendizaje de máquina, cuya parte forman las máquinas de vectores de soporte, elegidas para las tareas de clasificación. Luego, se transcriben distintas técnicas de procesamiento de señales discretas, necesarias para poder clasificar una señal. Se han elegido dos técnicas de extracción de características, siendo éstas la estadística de alto orden y las transformadas de Wavelet con base Daubechies 4 y Daubechies 6. A su vez, se han seleccionado las siete mejores técnicas halladas para implementar la reducción de las características, quedándose sólo con las más relevantes de todo el conjunto.

A continuación se elige la metodología para implementar el proyecto. El software elegido para implementarla se compone de dos toolbox, Libsvm y Feature Selections Algorithms. Posteriormente, se realiza una prueba sencilla de clasificación de tres pulsaciones, siendo los resultados obtenidos para clasificadores individuales One Against All (OAA) cercanos al cien por cien.

En la siguiente parte del proyecto se implementa la clasificación de ocho tipos de pulsaciones mediante los métodos mencionados anteriormente. Los resultados obtenidos son satisfactorios para todas las técnicas. El mejor método hallado ha sido basado en la estadística de alto orden y el análisis discriminante lineal, que se describe a continuación de forma detallada.

Finalmente, se presentan las conclusiones de este proyecto y las posibles líneas de una futura investigación, relacionadas con las clasificaciones de señales mediante máquinas de vectores de soporte.

Summary

A classification algorithm in the biomedicine field, particularly focused in the ECGs beats, is presented in this project. There are eight types of beats, composed of seven related to some diseases y one of the healthy person, to be classified. There is one additional difficulty in the classification tasks: two beats are composed of the fusion of another beats presented in the classification.

The project starts with the machine learning theoretical basis review. The support vector machine, used for the classification problems solutions, forms part of the machine learning. Also some discrete signal processing techniques, needful to be implemented before classification, are described. There are two techniques for feature extraction: high order statistics and Wavelet transform with Daubechies 4 and 6 bases. Other seven different algorithms are used for feature selection propose in way to preserve only the most relevant features.

In the next step the appropriate methodology and software are selected to meet the project objectives. Two Matlab toolbox are used for the software propose: Libsvm and Feature Selection Algorithms. Also the simple three classes' case classification was implemented with great results near a hundred percents.

After that, eight beats classification, using mentioned earlier features extraction and selection techniques, is implemented. The achieved results are satisfactory for all techniques. The best performing algorithm, that comes later with more details, is based on the high order statistics and linear discriminant analysis.

Finally, the project conclusions and possible future investigations issues in machine learning field are presented.

Acrónimos

En el presente proyecto se emplearán los siguientes acrónimos:

ANOVA	Análisis de varianza
CFS	Selección de características basada en correlación (Correlation based feature selection)
DB4	Daubechies 4
DB6	Daubechies 6
DWT	Transformada de Wavelet discreta (Discret Wavelet Transform)
ECG	Electrocardiograma
FDR	Ratio discriminante de Fisher (Fisher Discriminant Ratio)
GA	Algoritmo genético (Genetic Algorithm)
HOS	Estadística de alto orden (High Order Statistics)
InfoGain	Ganancia de la información
LDA	Análisis discriminante lineal (Lineal discriminant analysis)
OAA	Uno contra todos (One Against All)
RBF	Función de base radial (Radial basis function)
SARSA	Algoritmo de aprendizaje maquina por refuerzo State-Action-Rewarde-State-Action
SVM	Máquinas de vectores de soporte (Support Vector Machines)
VC	Dimensión de Vapnik-Chevronenkis
WT	Transformada de Wavelet (Wavelet transform)

Índice

RESUMEN	4
SUMMARY	5
ACRÓNIMOS	6
INDICE	7
1. INTRODUCCIÓN Y OBJETIVOS	9
1.1 Motivación	9
1.2 Objetivos y Metodología	10
1.3 Estructura del documento	11
2. TÉCNICAS DE PROCESAMIENTO Y CLASIFICACIÓN DE SEÑALES TEMPORALES	13
2.1 Introducción	13
2.2 Generalidades sobre aprendizaje	14
2.2.1 Teoría Estadística del Aprendizaje	14
2.2.2 Aprendizaje de Máquina	16
2.3 Comparación de las Máquinas de Vectores de Soporte con otras herramientas de aprendizaje de máquina	17
2.4 Máquinas de Vectores de Soporte (SVM)	20
2.4.1 Introducción a las SVM	20
2.4.2 Solución del problema de clasificación mediante SVM	21
2.5 Obtención de características de señales temporales	26
2.5.1 Estadística de Alto Orden (HOS)	27
2.5.2 Transformada de Wavelet (WT)	28
2.6 Reducción de características	31
2.6.1 Análisis Discriminante Lineal o el Ratio Discriminante de Fisher	31
2.6.2 Método Kruskal Wallis	32
2.6.3 Test Chi-Square	33
2.6.4 Índice de Gini	34
2.6.5 Ganancia de la Información (Information Gain)	34
2.6.6 Método de Mínima Redundancia y Máxima Relevancia	35
2.6.7 Selección de características basada en correlación (CFS)	35
3 TÉCNICAS DE CLASIFICACIÓN APLICADAS A LAS SEÑALES ECG	37
3.1 Introducción	37
3.2 Estado del arte	37
3.3 Base de datos y señales	38
3.4 Plataforma	39

3.5	Ejemplo: Clasificación de 3 clases	40
3.5.1	Caso HOS y LDA	42
3.5.2	Caso WT Db4 y Chi-Square	48
3.6	Conclusiones	49
4	IDENTIFICACIÓN DE PULSACIONES CARDÍACAS USANDO CLASIFICACIÓN SVM	50
4.1	Introducción	50
4.2	Sistema de Identificación de pulsaciones cardíacas propuesto	50
4.2.1	Ejemplos de tipos de arritmia	50
4.2.2	Obtención de modelos de clasificación binarios	52
4.2.3	Clasificador de pulsaciones basado en SVM	53
4.2.4	Comparación de resultados obtenidos aplicando diferentes técnicas de extracción y reducción de características	55
4.2.5	El mejor clasificador hallado: HOS y LDA	57
4.3	Conclusiones	60
5	CONCLUSIONES Y TRABAJO FUTURO	61
5.1	Conclusiones	61
5.2	Trabajo futuro	61
	BIBLIOGRAFÍA Y REFERENCIAS	62
	ANEXO A: “Cálculos de los parámetros de kernel mediante Libsvm”	66
	ANEXO B: “Otros cálculos de interés realizados en Matlab”	68

Capítulo 1: Introducción y objetivos

1.1 Motivación

Desde los años noventa, con la introducción masiva de los ordenadores y las redes de comunicaciones en todas las áreas de la actividad humana, la cantidad de información disponible ha aumentado de forma drástica. Dentro de este mar de información, hoy en día digitalizable y fácilmente procesable, a menudo hay ciertos datos de interés que son cruciales para tomar una decisión u otra.

Debido a la necesidad surgida de procesar gran cantidad de información, en los años noventa empezó el “boom” de los estudios de diferentes técnicas de procesamiento. La mayoría de las técnicas ya eran desarrolladas y conocidas décadas atrás (como es el caso de las máquinas de vectores de soporte (SVMs) de Vapnik de los años setenta [Vap 74], [Vap 79]), pero no se empleaban de forma genérica al no disponer de herramientas computacionales adecuadas.

Entre las técnicas relevantes de procesamiento de información cabe destacar las de toma de decisiones, que son a su vez las herramientas de clasificación. Dichas herramientas separan la información de interés en varios grupos previamente definidos.

Un ejemplo práctico de la aplicación de estas técnicas es la clasificación de pulsaciones de señales de electrocardiogramas (ECGs). Este tema ha sido de gran interés en los últimos años porque a la vez que permite profundizar en el diagnóstico automatizado de las enfermedades cardíacas proporciona una base experimental sólida para usarla en la clasificación de otras señales temporales.

Teniendo en cuenta que algunos pacientes requieren estudios cardíacos prolongados, con los registros correspondientes de los ECG de 24 a 48 horas, la interpretación requerida de señales producidas se convierte en una tarea laboriosa y monótona, requiriendo trabajo de personal altamente cualificado. Esta característica hace a las técnicas de interpretación de las ECGs muy atractivas, sobre todo a aquellas que se puedan implementar en equipos de dimensiones reducidas para su posterior uso tanto en los centros médicos como en los domicilios particulares de los pacientes que lo necesiten.

Para implementar un sistema de diagnóstico de señales cardíacas es preciso conocer las propiedades principales de un electrocardiograma. De forma genérica, un electrocardiograma consiste en la representación gráfica de la actividad eléctrica de corazón. Éste produce un patrón característico de variaciones de voltaje y se contrae cíclicamente al recibir impulsos eléctricos. Los componentes del corazón implicados en su correcta contracción son cuatro cámaras interconectadas entre sí mediante válvulas

y fibras del músculo cardíaco. Si algún componente presenta deficiencias, esto se traduce en cambios de las pulsaciones de los ECGs.

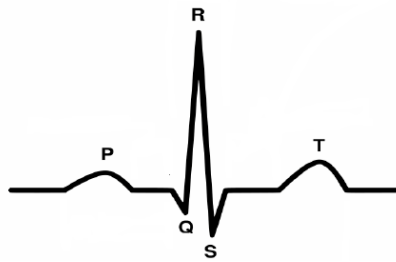


Fig. 1.1 Modelo del pulso cardíaco “normal” con sus principales elementos.

Una pulsación cardíaca normal (Fig. 1.1) consiste de una onda P, un complejo QRS y una onda T. Cada uno de estos elementos de la pulsación corresponde a una actividad eléctrica concreta. Cuando se produce alguna alteración importante de uno o varios elementos, la pulsación se considera anómala.

De esta manera, se puede detectar una anomalía cardíaca mediante un algoritmo de clasificación de pulsaciones, cuya implementación es el corazón de este proyecto.

1.2 Objetivos y Metodología

Tras existir un numeroso grupo de distintos trabajos de investigación en el campo de las ECGs ([Meh 07], [Jal 09], [Son 05], [Kar 10], [Yin 12], [Ozl 10], [Can 10] y [Can12]) sería de gran interés comparar diferentes técnicas de procesamiento de señales cardíacas con el fin de hallar el método más óptimo. Durante este estudio se pretende alcanzar dos metas importantes: adquirir experiencia en el campo de procesamiento y clasificación de señales, con posible futura aplicación en otras áreas de investigación, y la obtención de un algoritmo preciso y fiable de detección de anomalías cardíacas. Una vez obtenido este algoritmo, se abren las puertas a su posible implementación en hardware, siendo estos trabajos hoy en día muy actuales por las necesidades médicas presentes y la tecnología disponible para llevarlo a cabo.

Para abordar los objetivos planteados, se requieren tres pasos importantes, siendo éstos el estudio bibliográfico de procesamiento de ECGs, las pruebas y la comparación. Dichos pasos, representan tres pilares del proyecto y corresponden a la metodología del mismo.

En primer lugar, se parte de un estudio bibliográfico de las técnicas de procesamiento de ECGs empleadas por otros equipos de investigación. Se presta especial atención a las que abordan un grupo grande de anomalías, varios registros de datos y que proporcionan unos atractivos resultados.

Después de haber estudiado las diferentes técnicas de procesamiento que se suelen emplear, se generalizan las mismas, abordando un estudio matemático de forma genérica buscando otras alternativas.

Una parte del estudio bibliográfico cubre también el estudio del software necesario para implementar las técnicas de procesamiento elegidas.

Después de haber profundizado en las técnicas de procesamiento de ECGs y escogido los medios informáticos para implementarlas, se pasa a otra parte esencial del proyecto, en la que se llevan a cabo las pruebas de dichas técnicas.

La parte de procesamiento de un registro digitalizado de ECG empieza con la obtención de pulsaciones (en este proyecto se emplearán ocho diferentes clases de pulsaciones), aplicando a continuación técnicas de extracción de características. Una vez obtenidas las características se requiere reducirlas para el posterior uso en un algoritmo de clasificación, siendo el resultado final del mismo la identificación de la pulsación (si es normal o representa alguna anomalía en concreto).

Debido a que no existe una técnica específica y única para extraer o reducir las características de una señal, se abre un abanico de posibilidades a la hora de probar distintas combinaciones. Se empieza con una técnica de extracción, combinándola con diferentes métodos de reducción de características. Después, se repite lo mismo para otras técnicas de extracción.

Una vez finalizadas las pruebas, se realiza una comparación de resultados obtenidos. En primer lugar, se elige la técnica de extracción que ha dado mejores resultados, seleccionando a continuación una técnica de reducción. La mejor combinación hallada de estas técnicas, conjuntamente con el algoritmo de clasificación, representa el algoritmo de detección de anomalías que se pretendía encontrar. Siendo importante conocer los detalles de este algoritmo, se presentan todas sus particularidades como el resultado final de la fase de comparación.

1.3 Estructura del documento

La memoria del proyecto se compone de cinco capítulos. En el primero se describe la motivación del mismo, se definen los objetivos que se pretenden alcanzar y se elige la metodología a seguir. En el segundo capítulo se realiza un estudio bibliográfico de diferentes técnicas de procesamiento y clasificación de señales temporales, describiendo a la vez fundamentos matemáticos de las técnicas elegidas. En el tercer capítulo se estudia la aplicación de las técnicas de clasificación a las señales ECG, se describe la base de datos empleada y la plataforma elegida, realizándose una prueba de clasificación de tres clases.

En el siguiente capítulo se implementa la parte práctica del proyecto, proponiendo un sistema de identificación de pulsaciones cardíacas. Así mismo, se describen las pulsaciones elegidas y los modelos de clasificadores binarios obtenidos. También se hace un estudio comparativo de las técnicas de extracción y selección de características expuestas en el capítulo dos. La mejor combinación hallada se describe detalladamente.

Y finalmente, en el quinto y último capítulo, se presentan las conclusiones del proyecto y se elaboran propuestas para posibles futuros trabajos.

Capítulo 2: Técnicas de procesamiento y clasificación de señales temporales

2.1 Introducción

Una señal representa variaciones de una función o de una variable en el espacio o en el tiempo. Estas variaciones, en amplitud o frecuencia, son de gran interés porque contienen información que lleva incorporada la señal. Esta información puede ser crítica, como es el caso de señales de vibración de un transformador eléctrico con cambiador de tomas descrito en la referencia [Riv 09]. Analizando las vibraciones, grabadas en un registro, se puede detectar las averías del cambiador de tomas mediante un patrón típico.

Para poder llevar a cabo una correcta identificación de un patrón, es necesario acotar la señal o extraer de ella un área de interés, con la posterior obtención de las características. Las características de una señal se extraen mediante una de las siguientes formas:

- Algunos datos de interés conocidos: distancias, amplitudes, periodos, etc... Las desventajas de este enfoque consiste en que es necesario el conocimiento profundo de las propiedades de la señal, siendo la experiencia adquirida de poca utilidad para trabajar con otras señales. Por otro lado, esta técnica depende del factor humano, por lo que se pueden duplicar unos datos y perderse otros relevantes.
- Obteniendo una descomposición o una transformada de la señal. Posee la ventaja de poder emplearse con diferentes señales sin precisar conocimientos previos de sus propiedades relevantes. La técnica más conocida es la transformada de Wavelet que ofrece como resultado un análisis en tiempo-frecuencia mediante coeficientes de detalle y aproximación para distintos niveles de descomposición.

Una vez obtenidas las características, éstas a menudo tienen información redundante que dificulta una correcta búsqueda de un patrón o la identificación de una señal, al existir esta información en todos los grupos. Para facilitar el proceso de búsqueda o identificación es conveniente quedarse únicamente con las características imprescindibles sin que haya información irrelevante.

Un ejemplo de la búsqueda de un patrón es al caso de dos grupos, cuando uno contiene el patrón buscado y el otro no. Al haber solo dos grupos de datos, el proceso se llama binario. Los grupos de datos se nombran clases y el proceso que identifica un dato como una clase u otra se llama clasificación (en este caso binaria). Los modelos

matemáticos (normalmente problemas de optimización) empleados en el proceso de clasificación se llaman clasificadores y se obtienen a partir de datos históricos con el objetivo de identificar datos nuevos del mismo tipo.

El campo matemático que trata temas de clasificación se llama aprendizaje máquina, al dotar en cierta forma a las máquinas de inteligencia, al ser capaces de encontrar un patrón o identificar un dato de interés mediante modelos matemáticos.

En los siguientes capítulos se desarrollan técnicas de clasificación formando parte del aprendizaje máquina, se presentan las técnicas de extracción de características mediante la transformada de Wavelet o cumulantes y se describen distintas técnicas de reducción de características.

2.2 Generalidades sobre aprendizaje

2.2.1 Teoría Estadística del Aprendizaje

El avance en el ámbito del aprendizaje comenzó a mediados del siglo pasado, al publicarse en el año 1949 la teoría del aprendizaje neuronal por D.O. Hebb en su libro de la referencia [Heb 49] y, un año más tarde, la teoría estadística del aprendizaje por W.K. Estes [Est 50]. También cabe remarcar los trabajos relacionados con esta teoría de R.R. Bush y F. Mostelelr [Bus 55] publicados en el año 1955.

La teoría estadística del aprendizaje representa el campo de aprendizaje automático que se basa en la estadística y el análisis funcional. A su vez, el análisis funcional está relacionado con el estudio de espacios de funciones. Las raíces históricas de este análisis son los estudios de transformaciones, como la Transformada de Fourier, y estudios de ecuaciones diferenciales e integrales.

Inicialmente, el análisis funcional se atribuía al estudio de los espacios vectoriales normados (espacios de Banach) sobre los reales o los complejos. Un ejemplo de los espacios de Banach es el de Hibert, en cuyo caso la norma surge de un producto escalar. En la actualidad el análisis funcional está visto como el estudio de espacios de vectores, dotados de topología (cuyo caso particular son los espacios de dimensiones infinitas).

Volviendo a la teoría estadística del aprendizaje, cabe remarcar que ésta se centra en la búsqueda de una función predictiva basada en datos. Hoy en día estas técnicas de predicción se usan a menudo en muchos campos, como la bioinformática, visión artificial... etc

El proceso de aprendizaje puede ser de muchos tipos, siendo los más conocidos el aprendizaje supervisado y el no supervisado. Durante el aprendizaje supervisado se parte de un conjunto de datos de entrenamiento. Cada dato en este conjunto se considera como una entrada que luego se asocia con una salida. De esta forma, el proceso de aprendizaje consiste en encontrar una función (o un modelo predictivo) que asigne a cada entrada una salida. Posteriormente, esta función se usará para predecir las salidas a partir de nuevas entradas.

En función del tipo de entrada, el problema de aprendizaje supervisado se considera un caso de regresión o un caso de clasificación. Si la salida es continua, es un problema de regresión. De forma contraria, el problema de clasificación se da para unas salidas discretas.

Para representar el aprendizaje en formato matemático, se parte de un espacio vectorial X con todas las posibles entradas, correspondiendo a su vez el espacio vectorial Y a todas las posibles salidas. Según la teoría estadística del aprendizaje, tiene que haber una distribución de probabilidad desconocida $p(z) = p(\vec{x}, y)$ en el espacio producto $Z = X \otimes Y$.

Entonces, el conjunto de entrenamiento podría ser un numero n de muestras obtenidas de una probabilidad de distribución. En este caso, el producto Z se puede anotar de esta forma:

$$\{(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)\} = \{\vec{z}_1, \dots, \vec{z}_n\} \quad (2.1)$$

siendo \vec{x}_i un vector de entrada perteneciente al conjunto de entrenamiento, y y_i la salida correspondiente al vector de entrada.

De esta manera, el problema se reduce a la búsqueda de una función $f: X \rightarrow Y$, ó $f(\vec{x}) \sim y$ dentro de \mathcal{H} , un espacio hipotético de funciones $f: X \rightarrow Y$.

Por otro lado, habrá una diferencia $V(f(\vec{x}, y))$ entre el valor predicho $f(\vec{x})$ y el valor real, y .

El riesgo esperado se puede definir así:

$$I[f] = \int_{X \otimes Y} V(f(\vec{x}, y)) p(\vec{x}, y) d\vec{x} dy \quad (2.2)$$

Teniendo en cuenta la expresión del riesgo, la función f que se pretende encontrar es la que satisface lo siguiente:

$$\inf_{f \in \mathcal{H}} I[f] \quad (2.3)$$

Debido a que la distribución de probabilidad $p(\vec{x}, y)$ no se conoce, se hace una medida aproximada del riesgo utilizando el conjunto de datos de entrenamiento. Esta medida del riesgo se llama riesgo empírico y se define a continuación:

$$I_S[f] = \frac{1}{n} \sum_{i=1}^n V(f(\vec{x}_i), y_i) \quad (2.4)$$

El algoritmo que selecciona la función f_S que minimiza el riesgo empírico se conoce por el nombre de la “minimización del riesgo empírico”.

La función de pérdidas es relevante para que el algoritmo de aprendizaje seleccione una f_S concreta. El ratio de convergencia del algoritmo también se ve afectado por la función de pérdidas. Según [Ros 04] es importante que la función de perdidas sea convexa.

Para un algoritmo de clasificación se suele emplear la función de pérdidas 0-1. Esta función toma el valor de 0 si la salida predicha es la misma que la salida actual. De forma contraria, toma el de 1, si la salida predicha difiere de la actual. Para una clasificación binaria la función de perdidas 1-0 se puede expresar de la siguiente forma:

$$V(f(\vec{x}, y)) = \theta(-yf(\vec{x})) \quad (2.5)$$

donde θ significa función escalón de Heaviside. Debido a que la función de perdidas 0-1 no es convexa, ésta se utiliza a menudo en forma modificada:

$$V(f(\vec{x}, y)) = (-yf(\vec{x}))_+ \quad (2.6)$$

2.2.2 Aprendizaje de Máquina

El aprendizaje de máquina forma parte de la inteligencia artificial y se centra en los estudios de distintos sistemas capaces aprender a partir de datos. El núcleo de aprendizaje máquina consiste principalmente en la representación y la generalización. La parte de la representación corresponde al sistema de aprendizaje, que trata con datos en forma de entrada y funciones que evalúan estas entradas. A su vez, la generalización es la capacidad del sistema de obtener buen rendimiento incluso con entradas nuevas, desconocidas previamente.

En función del tipo de algoritmo, el aprendizaje de máquina se puede ubicar en uno de los siguientes seis grupos:

› **Aprendizaje Supervisado:** proporciona una función que obtiene salidas a partir de entradas. Las salidas a menudo son etiquetas, asignadas a los grupos de datos de entrenamiento.

Algunos algoritmos conocidos que pertenecen a este grupo son los siguientes: redes neuronales, clasificadores bayesiana, árboles de decisión, programación genética, aprendizaje del autómatas, algoritmo del vecino más cercano y las máquinas de vectores de soporte.

› **Aprendizaje No Supervisado:** modela unos grupos de entradas en forma de cúmulos. En este caso las etiquetas no se conocen durante el entrenamiento.

Algunos ejemplos de este tipo de aprendizaje son: las redes neuronales, algoritmo de agrupamiento de datos, mapa auto-organizado, redes con funciones de activación de base radial y el algoritmo de cuantificación vectorial.

› **Aprendizaje Semi- Supervisado:** combina las muestras de entrenamiento etiquetadas con las no etiquetadas con el fin de obtener una función de clasificación idónea.

› **Aprendizaje Por Refuerzo:** Su idea se centra en aprender a actuar simulando las interacciones del entorno real. Cualquier acción tiene un impacto en el entorno. El entorno, al percibir una acción, genera una respuesta que va guiando el algoritmo de aprendizaje llegando a éste por vía de retroalimentación.

Los ejemplos de este aprendizaje son los siguientes: aprendizaje autómatas, Q-aprendizaje, método de Monte Carlo, algoritmo SARSA (state-action-reward-state-action) y el aprendizaje de diferencia temporal.

› **Aprendiendo A Aprender:** se construyen vías de decisión en base a experiencia previa.

› **Aprendizaje Evolucionario:** se desarrolló para el aprendizaje de los robots. Durante este aprendizaje se generan las secuencias (llamadas a menudo curriculum) de las situaciones de aprendizaje para adquirir las destrezas a través de las exploraciones autónomas e interacciones sociales con profesores humanos.

2.3 Comparación de las Máquinas de Vectores de Soporte con otras herramientas de aprendizaje de máquina.

Las herramientas de aprendizaje de máquina más conocidas y utilizadas en la actualidad son Algoritmos Genéticos, Redes Neuronales y las Máquinas de Vectores de Soporte.

Los algoritmos genéticos (GA) son unos modelos computacionales inspirados en la evolución. Este método se utiliza para encontrar soluciones exactas o aproximadas para problemas de optimización o búsqueda. Es de utilidad extrema cuando se dispone de un gran número de variables y se pretende encontrar la mejor

combinación de estas variables. En estos algoritmos la solución a un problema se contiene en una estructura cromosómica. Una vez obtenida la solución, se aplican distintas operaciones a esta estructura, como reproducción, cruces y mutaciones con el fin de preservar información crítica.

La implementación del algoritmo genético comienza con una población (normalmente aleatoria) de cromosomas. Después se evalúan las estructuras y se ubican las estructuras reproductivas. De esta manera, el algoritmo trabaja codificando las variables, en vez de trabajar directamente con las variables en sí.

Algunas de las propiedades más llamativas del algoritmo genético son las siguientes:

- En caso de un algoritmo genético binario se trabaja con un espacio de búsqueda discreto, incluso si las funciones son continuas.
- Se pueden abordar en el algoritmo tanto las funciones discretas como continuas. Esta propiedad permite que el GA se puede emplear en muchos campos.
- También cabe destacar la ventaja del GA que consiste en el aprovechamiento de similitudes en estructuras tipo cadena para llevar a cabo búsquedas efectivas.
- Por otro lado, el mayor problema radica en que es necesario seleccionar previamente un código adecuado para que el GA dé buenos resultados. Es difícil saber a priori que código es el mejor. En el estudio [Pan 10] se recomienda el uso del código que respete los bloques constructivos subyacentes.
- Se emplea un tiempo elevado para encontrar una solución a un problema.
- Se recomienda emplear esta técnica cuando se puede evaluar la “calidad” de una solución.

Otro método de aprendizaje de máquina son las redes neuronales, que se utilizan para modelar las relaciones complejas entre entrada y salida o para encontrar algunos patrones en datos. Esta herramienta es bastante útil cuando se dispone de una gran cantidad de datos en varias clases y se pretende clasificar en el futuro muestras nuevas. Durante la implementación de una red neuronal se sigue el modelo simplificado de las interconexiones de neuronas humanas. Para una mejor predicción de una salida a partir de una entrada se emplea el factor de pesos.

Existe una gran familia de redes neuronales artificiales que van variando en función del número y tipo de capas (ocultas o visibles, de entrada o salida), de la direccionalidad de las conexiones de las neuronas y del tipo de aprendizaje (supervisado o no supervisado, competitivo o por refuerzo). La ventaja de esto consiste en que se puede seleccionar una red neuronal que proporcione mejores resultados a un problema particular.

El algoritmo más conocido y empleado de las redes neuronales es el de propagación hacia atrás (empleándolo en las capas alimentadas hacia delante). La idea principal según [Col 01] consiste en organizar las neuronas en capas, enviando las señales de estas neuronas hacia delante, propagándose luego los errores hacia atrás. Las entradas en esta red son recibidas por las neuronas en la capa de entrada y las salidas se dan por las neuronas ubicadas en las capas de salida. También puede haber más de una capa intermedia oculta. Para un caso supervisado, se va reduciendo el error durante la etapa de entrenamiento mediante el ajuste de los pesos de las neuronas.

En las redes neuronales hay una función, llamada función de activación, que es la encargada de activar o no una neurona, dada una o varias entradas. Para el caso de propagación hacia atrás, la función de propagación depende sólo de las entradas y de los pesos, definiéndose a continuación:

$$A_j(\bar{x}, \bar{w}) = \sum_{i=0}^n x_i w_{ji} \quad (2.7)$$

Para las neuronas lineales la función de salida es la misma que la de activación. Normalmente las neuronas no son lineales y su función de salida es sigmoideal:

$$O_j(\bar{x}, \bar{w}) = \frac{1}{1 + e^{A(\bar{x}, \bar{w})}} \quad (2.8)$$

La función sigmoideal se aproxima a uno para valores positivos grandes y, se aproxima a cero para valores negativos. Igualmente, como la función de activación, la de salida depende solo de las entradas y los pesos.

Cabe remarcar que el error, siendo igual a la diferencia entre la salida real y la deseada, depende de los pesos. Entonces, para minimizar el error es necesario ajustar los pesos. Para este ajuste se emplea una función que mide la eficiencia. Esta función, en la mayoría de los casos, es una función compleja que presenta muchos mínimos locales. El mejor conjunto de pesos es aquel que hace que la función alcance un mínimo local.

Todo lo anterior hace que la red neuronal presenta las siguientes desventajas:

- Es difícil encontrar un conjunto de pesos para que la eficiencia sea máxima.
- El entrenamiento es costoso en términos computacionales.

Centrándose en las ventajas de las redes neuronales, se pueden citar las siguientes:

- Las capas ocultas permiten trabajar con espacios de cualquier dimensión.
- La clasificación es muy eficiente.
- Se puede diseñar para cada caso particular el número de capas ocultas y nodos.

Otra técnica de aprendizaje máquina que se describirá a continuación, son máquinas de vectores de soporte. Esta técnica típicamente se emplea para el reconocimiento de patrones mediante la búsqueda de un hiperplano de separación entre dos clases de datos. El hiperplano se obtiene buscando solución a un problema de programación cuadrática, dependiente del parámetro de regularización. Los datos que se pretende separar se pasan al espacio con mayor número de dimensiones mediante el truco de kernel, con el fin de facilitar la separación de datos. Todo esto hace que esta técnica presente las siguientes propiedades:

- El espacio de búsqueda sólo tiene un mínimo local, por lo que la solución del problema de optimización es única.
- Los kernels permiten trabajar en espacios de mayores dimensiones, facilitando la obtención del hiperplano.
- El entrenamiento y la clasificación son muy eficientes.
- Se puede especificar para cada caso en concreto el tipo de kernel y el valor del parámetro de coste C .

2.4 Máquinas de Vectores de Soporte

2.4.1 Introducción a las SVM

La técnica de las máquinas de vectores de soporte, introducida a mediados del siglo pasado y, finalmente diseñada en su empleo actual (mediante el margen blando) por Vapnik en el año 1995, trata de minimizar el riesgo estructural. De forma contraria, otras técnicas de reconocimiento de patrones se centran en la minimización del riesgo empírico.

El enfoque del riesgo empírico según [Pon 97] pretende optimizar el rendimiento del conjunto de entrenamiento, mientras que el riesgo estructural está enfocado en la probabilidad de los errores de clasificación para una distribución desconocida de datos. El riesgo estructural equivale a la minimización del error en su “capa superior”, siendo esto posible gracias a la teoría de la convergencia uniforme de probabilidad [Pon 97].

La SVM presenta la habilidad de poder condensar la información contenida en el conjunto de entrenamiento. Por otro lado, cabe mencionar que las familias de superficies de decisión presentan una dimensión VC (dimensión de Vapnik-Chervonenkis, de forma genérica es $n+1$, donde n es el número de dimensiones) relativamente baja.

Para un caso linealmente separable, dado un conjunto de entrenamiento S con dos clases de muestras, éstas se separan mediante un hiperplano determinado por algunos puntos S , llamados vectores de soporte.

Para el caso separable, el hiperplano maximiza el margen, que es dos veces la distancia desde el hiperplano a cada clase. En la práctica, dos clases en la mayoría de veces no se separan por completo, siendo necesario plantear un problema de optimización. El problema de optimización en este caso busca un compromiso entre el máximo margen posible y los mínimos errores posibles, controlados mediante un parámetro de regularización o coste C .

2.4.2 Solución del problema de clasificación mediante SVM

En primer lugar se define el conjunto S de datos $x_i \in \mathbb{R}^n$, con $i = 1, 2, \dots, N$. Cada punto x_i pertenece a una de dos clases: 1 o -1. Las etiquetas de x_i están dadas en $y_i \in \{-1, 1\}$.

La solución del problema de clasificación se da mediante una ecuación de hiperplano que divide S dejando los puntos de cada clase en un lado diferente del hiperplano, de forma que la distancia entre el hiperplano y cada clase sea la máxima posible.

Caso linealmente separable:

El conjunto S es linealmente separable cuando existe $w \in \mathbb{R}^n$ y $b \in \mathbb{R}$ que satisfacen lo siguiente:

$$w \cdot x_i + b \geq 1 \quad \text{para } y_i = 1$$

$$w \cdot x_i + b \leq -1 \quad \text{para } y_i = -1$$

Reescribiendo las ecuaciones anteriores en forma más compacta, se quedan así:

$$y_i(w \cdot x_i + b) \geq 1 \quad (2.9)$$

Para cada caso de $i = 1, 2, \dots, N$ el hiperplano de separación se define por w y b :

$$w \cdot x + b = 0 \quad (2.10)$$

Siendo ω la norma de w , la distancia desde el hiperplano hasta x_i es la siguiente:

$$d_i = \frac{w \cdot x_i + b}{\omega} \quad (2.11)$$

Teniendo en cuenta las ecuaciones (2.10) y (2.11), para todas x_i pertenecientes a S se cumple que:

$$y_i d_i \geq \frac{1}{\omega} \quad (2.12)$$

donde $\frac{1}{\omega}$ es la distancia desde el hiperplano hasta el punto más cercano de S .

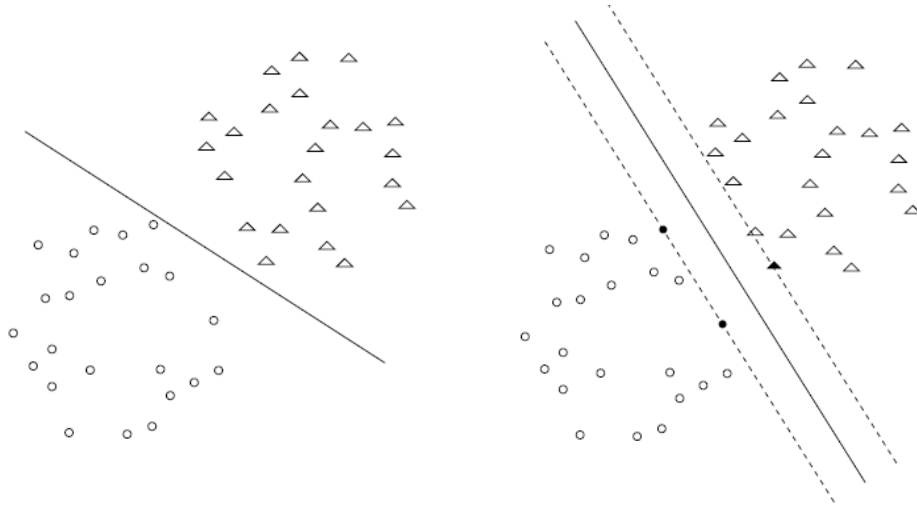


Fig.2.1 Un hiperplano de separación a la izquierda que no es óptimo, y el óptimo a la derecha.

Dado que el hiperplano de separación óptimo maximiza la distancia entre sí mismo y el punto más cercano de S , siendo esta $\frac{1}{\omega}$, la solución para obtener el hiperplano óptimo consiste en resolver un problema de optimización:

Problema de Optimización 1

$$\min \frac{1}{2} w \cdot w$$

$$\text{sujeto a : } y_i(w \cdot x_i + b) \geq 1 \quad i = 1, 2, \dots, N$$

Este problema se puede resolver mediante el método de multiplicadores de Lagrange. Denotando $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N)$ que son N multiplicadores no negativos (positivos o cero) de Lagrange, la solución del problema de optimización se traslada a la búsqueda del punto soporte de la siguiente función:

$$L = \frac{1}{2} w \cdot w - \sum_{i=1}^N \alpha_i \{y_i(w \cdot x_i + b) - 1\} \quad \text{con } L = L(w, b, \alpha) \quad (2.13)$$

En el punto de soporte L se tiene el mínimo para $w = \bar{w}$ y $b = \bar{b}$, y el máximo para $\alpha = \bar{\alpha}$.

Teniendo en cuenta lo anterior, esto equivaldría a

$$\frac{\partial L}{\partial b} = \sum_{i=1}^N y_i \alpha_i = 0 \quad y \quad \frac{\partial L}{\partial w} = w - \sum_{i=1}^N \alpha_i y_i x_i = 0 \quad (2.14)$$

$$\text{con } \frac{\partial L}{\partial w} = \left(\frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, \dots, \frac{\partial L}{\partial w_N} \right)$$

Todo esto hace que el problema de optimización anteriormente planteado se reduzca a un problema de maximización de la siguiente función:

$$l(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j x_i \cdot x_j, \text{ sujeto a } \alpha \geq 0 \quad (2.15)$$

El nuevo problema se llama el problema dual y se puede reformular así:

Problema de Optimización 2

Max $-\frac{1}{2} \alpha \cdot D \alpha + \sum \alpha_i$, sujeto a $\sum y_i \alpha_i = 0$, $\alpha \geq 0$, siendo ambas sumas para $i=1,2,\dots,N$

D – es la matriz NxN: $D_{ij} = y_i y_j x_i \cdot x_j$

A su vez, se puede obtener \bar{w} y \bar{b} a partir de las siguientes ecuaciones:

$$\bar{w} = \sum_{i=1}^N \bar{\alpha}_i y_i x_i,$$

$$\bar{\alpha}_i (y_i (\bar{w} \cdot x_i + \bar{b}) - 1) = 0, \quad i = 1, 2, \dots, N \quad (\text{Condición de Kuhn-Tucker}) \quad (2.16)$$

Donde x_i son vectores de soporte (puntos más próximos al hiperplano de separación).

Entonces, dado un vector de soporte x_j , se puede hallar \bar{b} mediante las condiciones de Kuhn-Tucker:

$$\bar{b} = y_j - \bar{w} \cdot x_j \quad (2.17)$$

De esta manera, los vectores de soporte condensan toda la información obtenida en el conjunto de entrenamiento S que es necesaria para poder clasificar datos nuevos.

El problema de clasificación se resuelve calculando la siguiente expresión para cada dato nuevo:

$$\text{sign}(\bar{w} \cdot x + \bar{b}) \quad (2.18)$$

Caso linealmente no separable:

Si un conjunto de datos S no es separable linealmente, no tiene ningún sentido buscar un hiperplano de separación óptimo porque puede darse el caso de que no haya ninguno.

No obstante, el análisis del caso lineal se puede generalizar añadiendo N variables no negativas $\xi = (\xi_1, \xi_2, \dots, \xi_N)$:

$$y_i = (w \cdot x_i + b) \geq 1 - \xi_i \quad i = 1, 2, \dots, N \quad (2.19)$$

En esta ecuación la variable ξ es nula para los casos linealmente separables. Para los demás casos es necesario resolver el siguiente problema de optimización:

Problema de Optimización 3

$$\min \frac{1}{2} w \cdot w + C \sum \xi_i$$

$$\text{sujeto a } y_i(w \cdot x_i + b) \geq 1 - \xi_i \quad i = 1, 2, \dots, N \quad \xi \geq 0$$

En este problema el término $C \sum \xi_i$ se puede ver como la medida de los errores durante la clasificación. A su vez, el parámetro C se puede ver como el parámetro de regularización. El hiperplano de separación óptimo tiende a maximizar la distancia mínima $\frac{1}{w}$ para valores bajos de C . De forma contraria, para valores altos de C se pretende minimizar el número de muestras clasificadas mal.

El problema dual del caso linealmente no separable viene a continuación.

Problema de Optimización 4

$$\text{Max } -\frac{1}{2} \alpha \cdot D \alpha + \sum \alpha_i$$

$$\text{Sujeto a } \sum y_i \alpha_i = 0$$

$$0 \leq \alpha_i \leq C \quad i = 1, 2, \dots, N$$

La matriz D es la misma que en caso linealmente separable. Para los casos de C suficientemente grande este problema de optimización se puede simplificar resolviendo el caso como linealmente separable.

Volviendo al problema 4, se obtienen los valores de \bar{w} y \bar{b} a partir de las siguientes expresiones:

$$\bar{w} = \sum_{i=1}^N \bar{\alpha}_i y_i x_i \quad (2.20)$$

$$\text{condiciones de Kuhn-Tucker } \begin{cases} \bar{\alpha}_i (y_i (\bar{w} \cdot x_i + \bar{b}) - 1 + \bar{\xi}_i) = 0 \\ (C - \bar{\alpha}_i) \bar{\xi}_i = 0 \end{cases} \quad (2.21)$$

En las ecuaciones (2.21) correspondientes a las condiciones de Kuhn-Tucker los valores de $\bar{\xi}_i$ corresponden a los valores de ξ_i en los puntos de ensilladura (saddle points). Igualmente que en el caso separable, los x_i para los cuales $\bar{\alpha}_i > 0$ son los vectores de soporte.

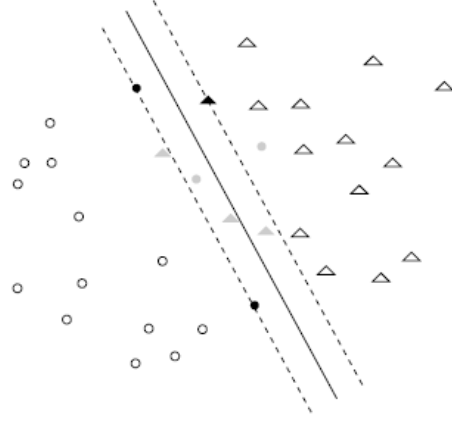


Fig. 2.2 Los vectores de margen en color negro y los errores en gris.

Cabe diferenciar entre dos grupos de vectores de soporte. En primer caso $\bar{\alpha}_i < C$, por lo que $\bar{\xi}_i = 0$ y los vectores se sitúan a una distancia $\frac{1}{\bar{\omega}}$ desde el hiperplano de separación óptimo. Estos vectores se llaman los vectores de margen.

Los vectores en el segundo caso, para $\bar{\alpha}_i = C$, están clasificados mal si $\xi_i > 1$, o, correctamente clasificados pero más cerca al hiperplano que $\frac{1}{\bar{\omega}}$ en caso de que $0 < \xi \leq 1$. En algunos casos, incluso estos vectores se ubican en las líneas de margen cuando $\xi_i = 1$. En cualquier suposición los vectores de soporte para los que $\bar{\alpha}_i = C$ se consideran errores.

Debido a que en la mayoría de los casos la separación lineal es imposible, se pueden buscar unas superficies de separación no lineales proyectando los puntos desde el espacio de entrada a un espacio de características con el fin de hallar un hiperplano óptimo en este nuevo espacio de características.

Si $x \in \mathbb{R}^n$ es un punto de entrada, entonces $\varphi(x)$ es el punto característico correspondiente, siendo φ (denominando con φ_i sus componentes) la proyección de x desde \mathbb{R}^n a un espacio Z (normalmente espacio Hilbert de dimensiones finitas o infinitas). Cabe destacar que a un hiperplano de separación óptimo en Z le corresponde un espacio de separación no lineal en el espacio de entrada \mathbb{R}^n .

Otra característica importante radica en que φ sólo se implicaría en los productos escalares de los puntos característicos:

$$D_{ij} = y_i y_j \varphi(x_i) \cdot \varphi(x_j) \quad \gamma \quad \bar{w} \cdot \varphi(x) + \bar{b} = \sum \bar{\alpha}_i y_i \varphi(x_i) \cdot \varphi(x) + \bar{b} \quad (2.22)$$

Esto hace que no sea necesario conocer por completo φ , ya que bastaría con encontrar una expresión del producto escalar en el espacio de características que emplea solamente los puntos de entrada:

$$\varphi(x_i) \cdot \varphi(x_j) = K(x_i, x_j) \quad (2.23)$$

Esta última expresión (2.23) se denomina **kernel** y se emplea para resolver el problema de optimización 4:

$$D_{ij} = y_i y_j K(x_i, x_j) \quad \text{sign}(\sum \bar{\alpha}_i y_i K(x_i, x) + \bar{b}) \quad (2.24)$$

De esta manera, la búsqueda de una solución para el caso linealmente no separable se reduce a un problema lineal dónde además es necesario hallar un kernel perteneciente a cierta familia de superficies de decisión

Una función $K(x, y) = \varphi(x) \cdot \varphi(y)$ con $x, y \in \mathbb{R}^n$ puede emplearse como un kernel si cumple el teorema de Mercer:

$$K(x, y) = K(y, x) \quad y \quad \iint K(x, y) f(x) f(y) dx dy > 0, \quad \forall f \in L^2 \quad (2.25)$$

A continuación vienen dos ejemplos de kernels.

En primer caso es el kernel polinómico:

$$K(x, y) = (1 + x \cdot y)^d \quad x, y \in [-a, a]^d \quad (2.26)$$

Este kernel satisface el teorema de Mercer y es finito. La superficie de separación en el espacio de entrada es de grado d .

El segundo ejemplo es el kernel Gaussiano RBF (Radial Basis Function):

$$K(x, y) = \exp\left(\frac{-\|x-y\|^2}{2\sigma^2}\right), \quad \sigma \in \mathbb{R} \quad (2.27)$$

Este kernel satisface el teorema de Mercer y es infinito.

2.5 Obtención de características de señales temporales.

En este apartado se definen las bases matemáticas a aplicar durante la extracción de características de un tramo de una señal digitalizada, en concreto de una pulsación cardíaca. No obstante, las técnicas descritas en 2.5.1 y 2.5.2 se pueden aplicar a otras señales digitalizadas cuando éstas se pueden acotar o descomponer en uno o varios tramos de interés.

2.5.1 Estadística de Alto Orden (HOS)

La estadística de alto orden, o High Order Spectral Analysis (HOSA) según su denominación en inglés, tiene sus raíces en la descomposición de las señales en sus componentes armónicos que empezó a emplearse para los estudios planetarios en el siglo diecisiete. No obstante, hasta finales del siglo pasado no se desarrolló una base matemática de la HOS, dando luego lugar a la aparición de las aplicaciones basadas en la estadística de alto orden.

Hoy en día las aplicaciones que utilizan esta técnica se pueden encontrar según [Men 91] en los campos de la biomedicina, física del plasma, procesamiento de los datos sísmicos, los datos de radar o sonar, filtros adaptivos, procesamiento de los arrays, reconstrucción de las imágenes etc...

Los términos más conocidos de las estadísticas de alto orden son cumulantes y sus transformadas de Fourier asociadas, llamadas polispectros. Tanto los cumulantes como los polispectros no sólo proporcionan la información relacionada con las amplitudes, sino además revelan la información relacionada con la fase. Esta característica es de suma importancia debido a que la estadística de segundo orden (por ejemplo las correlaciones) es ciega con respecto a la información de fase.

No obstante, los cumulantes no son sensibles a cualquier proceso Gaussiano, mientras que las correlaciones actúan de forma contraria. Debido a esta propiedad los métodos de procesamiento de las señales basados en cumulantes no dejan pasar el ruido Gaussiano de forma automática, mientras que los métodos basados en la correlación dejan pasar este ruido.

Teniendo en cuenta lo anterior, la estadística de alto orden es aplicable cuando se trabaja con procesos no Gaussianos (posiblemente no lineales), que es el caso de la mayoría de aplicaciones en distintas áreas científicas.

Normalmente se trabaja con tres cumulantes (que son espectros de alto orden):

- Espectro de potencia de segundo orden
- Espectro de tercer orden que es biespectro
- Espectro de cuarto orden que es trispectro

Las definiciones de los cumulantes de segundo y cuarto orden empleados para la extracción de características mediante de acuerdo a [Swa 98] vienen a continuación:

$$\hat{C}_{xy}(k) = \frac{1}{N_3} \sum_{n=N_1}^{N_2} x^*(n)y(n+k) \quad (2.28)$$

$$\hat{C}_{wxyz}(k, l, m) = \frac{1}{N_3} \sum_{n=N_1}^{N_2} w^*(n)x(n+k)y(n+l)z^*(n+m) - \\ C_{wxk}C_{yzl-m} - C_{wyl}C_{xzk-m} \quad (2.29)$$

Donde $N_1=0$, $N_2=N-1$, $N_3=N$, símbolo * representa conjugado y C_{xx} es cross-cumulante.

A su vez para el caso de autocumulantes $x=y=z=w$, y la dimensionalidad de cumulante de cuarto orden es 1.

Teniendo en cuenta lo anterior, las ecuaciones (2.28) y (2.29) se pueden resumir a lo siguiente:

$$\hat{C}_{2,x}(k) = \frac{1}{N} \sum_{n=0}^{N-1} x^*(n)x(n+k) \quad (2.28')$$

$$\hat{C}_{4,x}(k) = \frac{1}{N} \sum_{n=0}^{N-1} x^*(n)x(n+k)x(n+0)x^*(n+0) - \hat{C}_{2,x}(k) C_{2,x}(0 - \\ 0 - C_{2,x} C_{2,x} k - 0 = 1 N n = 0 N - 1 x^* n x^* n x n x(n+k) = 1 N n = 0 N - 1 (x^*(n)) \\ 2 x n x(n+k) \quad (2.29')$$

Donde $k=1,2,...,N-1$.

2.5.2 Transformada de Wavelet (WT)

De manera general, la Transformada Wavelet de una función $f(t)$ es la descomposición de $f(t)$ en un conjunto de funciones $\psi_{s,\tau}(t)$, que forman una base y son llamadas las "Wavelets".

$$W_f(s, \tau) = \int f(t) \psi_{s,\tau}^*(t) dt \quad (2.30)$$

siendo el significado del símbolo * complejo conjugado.

Las Wavelets son generadas a partir de la traslación y el cambio de escala de una misma función wavelet $\psi(t)$, llamada la "Wavelet madre", y se define como:

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-\tau}{s}\right) \quad (2.31)$$

siendo s el factor de escala, y τ el factor de traslación.

Las Wavelets $\psi_{s,\tau}(t)$ obtenidas de la misma función Wavelet madre $\psi(t)$ tienen diferente escala s y ubicación τ , pero tienen todas la misma forma. Se utilizan siempre factores de escala $s > 0$. Las Wavelets son dilatadas cuando la escala $s > 1$, y son contraídas cuando $s < 1$. De esta forma, cambiando el valor s se cubren rangos diferentes de frecuencia.

La Transformada de Wavelet se puede implementar sobre numerosas bases. Las diferentes categorías de Wavelets (continuas, discretas y ortogonales) y los varios tipos de funciones wavelets dentro de cada categoría proveen una gran cantidad de opciones para analizar una señal de interés. De esta manera, se puede elegir la base de funciones cuya forma se aproxime mejor a las características de la señal de interés.

Debido a que la forma de una pulsación se asemeja bastante a la forma de la base Daubechies 4 (Db4) según [Nir 13] o Daubechies 6 (Db6) según [Mah 05], se trabajará con estas Wavelets. Las bases Daubechies tienen la propiedad de formar una base orto normal, y debido a esto, se asegura la independencia de la representación de la señal en los diferentes niveles de descomposición, sin que se genere información redundante.

Por otro lado, las bases Daubechies permiten calcular la Transformada Wavelet mediante un algoritmo no muy complejo, con poco tiempo de cómputo, al contrario que las bases no ortonormales.

En el cálculo práctico de la Transformada de Wavelet orto normal, mediante base Daubechies, se utiliza un conjunto de filtros discretos paso-bajo (p_n) y paso-alto (q_n). Estos filtros son periódicos, de soporte compacto con un número finito N (siendo el grado de los filtros $N-1$) de coeficientes distinto de cero. De esta manera, los dos conjuntos de filtros forman una matriz de $2 \times N$ que posteriormente se utiliza en el algoritmo de descomposición:

$$p(0) \ p(1) \ \dots \ p(N-1)$$

$$q(0) \ q(1) \ \dots \ q(N-1)$$

Teniendo en cuenta la propiedad de la ortogonalidad de los filtros, se puede obtener el paso alto $q(n)$ a partir del paso bajo $p(n)$:

$$q(n) = (-1)^n p(N-1-n) \quad (2.32)$$

Sustituyendo el filtro $q(n)$ en la matriz por (2.32), ésta queda de la siguiente manera:

$$\begin{array}{ccccccc} p(0) & p(1) & \dots & p(N-1) & & & \\ p(N-1) & -p(N-2) & \dots & p(0) & & & \end{array} \quad (2.33)$$

La matriz resultante (2.33) se llama la “Matriz de Transformada Wavelet”. Esta matriz se emplea en el algoritmo de descomposición con el fin de obtener la matriz final.

A su vez, para una base Daubechies los coeficientes de filtro cumplen las siguientes relaciones algebraicas:

$$\sum_{m=0}^{N-1} p(m) = \sqrt{2} \quad (2.34)$$

$$\sum_{m=0}^{N-1} p(m)p^*(m+2k) = \delta(k,0) = \begin{cases} \sum_{m=0}^{N-1} |p(m)|^2 & \text{si } k = 0 \\ 0 & \text{si } k \neq 0 \end{cases} \quad (2.35)$$

$$\sum_{m=0}^{N-1} (-1)^m m^k p(N-1-m) = 0 \quad 0 \leq k \leq p-1 \quad (2.36)$$

donde $p = \frac{N}{2}$.

Trabajando con la base Daubechies es necesario tener en cuenta que su número (Db4, Db6, etc...) indica el número de coeficientes distintos de cero. La función DB4 es la más simple, teniendo sólo cuatro coeficientes no nulos. Por otro lado, cabe mencionar que los valores de la base Daubechies no están dados en forma analítica. Por ejemplo, los valores correspondientes a Db4 (N=4, p=2) son los siguientes:

$$p(0) = \frac{1}{4} \frac{(1 + \sqrt{3})}{\sqrt{2}} = 0,483$$

$$p(1) = \frac{1}{4} \frac{(3 + \sqrt{3})}{\sqrt{2}} = 0,836$$

$$p(2) = \frac{1}{4} \frac{(3 - \sqrt{3})}{\sqrt{2}} = 0,224$$

$$p(3) = \frac{1}{4} \frac{(1 - \sqrt{3})}{\sqrt{2}} = -0,13$$

Para un vector de entrada de tamaño L es necesario generar una matriz de Transformada Wavelet con los filtros de traslación discretos de tamaño LxL. Esta matriz se obtiene mediante las repeticiones de la matriz (3). Las filas impares corresponden a los filtros paso bajo p(n). En estas filas los filtros p(n) son trasladados dos columnas con respecto a la fila impar anterior. Lo mismo sucede con las columnas pares, a las que corresponden los filtros paso alto q(n).

$$\begin{pmatrix} c(1) \\ d(1) \\ c(2) \\ d(2) \\ . \\ . \\ . \\ . \\ . \\ . \end{pmatrix} = \begin{pmatrix} p(0) & p(1) & p(2) & p(3) & . & . & . & . & . & . & . \\ p(3) & -p(2) & p(1) & -p(0) & . & . & . & . & . & . & . \\ . & . & p(0) & p(1) & p(2) & p(3) & . & . & . & . & . \\ . & . & p(3) & -p(2) & p(1) & -p(0) & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & p(0) & p(1) & p(2) & p(3) \\ . & . & . & . & . & . & . & . & p(3) & -p(2) & p(1) & -p(0) \\ . & p(2) & p(3) & . & . & . & . & . & . & . & p(0) & p(1) \\ . & p(1) & -p(0) & . & . & . & . & . & . & . & p(3) & -p(2) \end{pmatrix} \begin{pmatrix} f(1) \\ f(2) \\ f(3) \\ f(4) \\ . \\ . \\ . \\ . \\ . \\ . \\ f(L) \end{pmatrix}$$

Fig. 2.3 Matriz Transformada de Wavelet para un vector $f(n)$.

Luego, al multiplicar el vector de datos $f(n)$ por la matriz Transformada de Wavelet (Fig. 2.3) se obtiene como resultado un vector con los coeficientes $c(n)$ y $d(n)$ intercalados. El vector $c(n)$ representa la aproximación discreta, siendo ésta la correlación entre el vector de datos $f(n)$ y los filtros $p(n)$. El otro resultado, $d(n)$, es la correlación entre $f(n)$ y los filtros $q(n)$, representando los coeficientes wavelet discretos.

De esta manera, la descomposición wavelet discreta se calcula aplicando la matriz de la Transformada Wavelet primero al vector $f(n)$ y luego, en cada iteración, a los coeficientes de aproximación $c(n)$. Como en cada iteración el vector de salida queda compuesto por los coeficientes $c(n)$ y $d(n)$ de forma intercalada, es necesario realizar una permutación de éste, para obtener en la primera mitad del vector de salida sólo los valores $c(n)$ y en la segunda mitad sólo los valores $d(n)$.

El método que se va a emplear para obtener las características en este apartado es el siguiente:

- 1) Aplicar la DWT (Transformada de Wavelet Discreta a cada conjunto normalizado de datos).
- 2) Obtener los coeficientes de aproximación del nivel de descomposición. Los coeficientes de detalles se descartan.

2.6 Reducción de Características.

De forma semejante que en 2.4 se definen a continuación las bases matemáticas de las técnicas a seguir durante la reducción de características.

2.6.1 Análisis Discriminante Lineal o el Ratio Discriminante de Fisher (LDA o FDR)

El Análisis Discriminante Lineal consiste en encontrar una combinación lineal de características que pueden caracterizar o separar dos o más clases. La combinación

resultante de características normalmente se suele usar para los propósitos de la reducción de la dimensionalidad previamente a una clasificación.

Este análisis está muy relacionado con el análisis de la varianza (ANOVA) y el análisis de regresión. En todas estas técnicas se pretende expresar una variable dependiente como una combinación lineal de otras variables, expresándose luego esta relación mediante un valor numérico.

En la práctica el análisis discriminante lineal se implementa mediante el ratio discriminante de Fisher, cuya definición viene a continuación.

Durante esta técnica según [Fer 09] los vectores de datos se proyectan a un subespacio resolviendo un problema de eigendescomposición. El subespacio se calcula de forma supervisada.

Para un caso de dos clases, el objetivo consiste en encontrar una dirección única \mathbf{w} , tal que las respectivas proyecciones de los vectores de características de l -dimensiones $\mathbf{x} \in \mathbf{R}^l$ maximicen el ratio discriminante de Fisher.

El ratio de Fisher (FDR) para una característica escalar y en caso de una clasificación binaria se puede definir de la siguiente manera:

$$FDR = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \quad (2.37)$$

donde μ_1 y μ_2 son valores medios de y , así como σ_1^2 y σ_2^2 son varianzas de y en ambas clases respectivamente.

El objetivo del FDR en caso de dos clases se puede resumir como sigue:

- que las varianzas de ambas clases sean las menores posible
- que el valor medio de una clase esté alejado lo más posible del valor medio de otra clase.

2.6.2 Método Kruskal Wallis.

Este método es no paramétrico (no hay ninguna suposición sobre la distribución de muestras) y se basa en la comparación de las medianas de los grupos a clasificar. Las muestras se ordenan en función de su valor (de menor a mayor).

Durante el primer paso se da un valor a todas las muestras recorriendo distintos grupos. La fórmula que se emplea para este cálculo es la siguiente:

$$K = (N - 1) \frac{\sum_{i=1}^g n_i (\bar{r}_i - \bar{r})^2}{\sum_{i=1}^g \sum_{j=1}^{n_i} (r_{ij} - \bar{r})^2} \quad (2.38)$$

Donde:

N – es el número de muestras en todos los grupos

n_i – es el número de muestras en el grupo i

r_{ij} - es la valoración (Kruskal Wallis) de la muestra j en el grupo i .

$$\bar{r}_i = \frac{\sum_{j=1}^{n_i} r_{ij}}{n_i} \quad (2.39)$$

$$r \text{ – es la media de todas las muestras, } r = \frac{N(N+1)}{2N} = \frac{N+1}{2} \quad (2.40)$$

Teniendo en cuenta que el denominador de K se puede simplificar de la siguiente manera:

$$\begin{aligned} \sum_{i=1}^g \sum_{j=1}^{n_i} (r_{ij} - \bar{r})^2 &= (r_{11} - \bar{r})^2 + (r_{12} - \bar{r})^2 + \dots + (r_{gn_i} - \bar{r})^2 = \\ &= \frac{(N(N+1)(2N+1))}{6} + ((\frac{(N+1)}{2})^2 * N) - \left(2 * \left(\frac{N+1}{2} \right) * \left(N * \frac{N+1}{2} \right) \right) = \frac{N(N+1)(N-1)}{12}, \end{aligned} \quad (2.41)$$

la fórmula (2.38) se puede transcribir:

$$K = \frac{12}{N(N+1)} \sum_{i=1}^g n_i \left(\bar{r}_i - \frac{N+1}{2} \right)^2 = \frac{12}{N(N+1)} \sum_{i=1}^g n_i (\bar{r}_i)^2 - 3(N+1)$$

2.6.3 Test Chi-Square

Durante el test Chi-Square se realizan dos comprobaciones: una de calidad de las muestras y otra de sus independencias. Durante la selección de características se comprueba si alguna clase en concreto es independiente de una muestra particular.

La implementación matemática de test Chi-Square para una característica con r distintos valores y C clases se define de la siguiente forma:

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^C \frac{(n_{ij} - \mu_{ij})^2}{\mu_{ij}} \quad (2.42)$$

donde n_{ij} es el numero de muestras con el valor i^{th} de una característica

y donde $\mu_{ij} = \frac{n_{*j}n_{i*}}{n}$, siendo n_{i*} numero de muestras con el valor i^{th} de una característica particular, y siendo n_{*j} numero de muestras en clase j , y siendo n numero de muestras.

2.6.4 Índice de Gini

El índice de Gini calcula la habilidad de una característica para poder distinguir clases. Dadas C clases, el índice se puede calcular de esta forma:

$$GiniIndex(f) = 1 - \sum_{i=1}^C [p(i|f)]^2 \quad (2.43)$$

Para una clasificación binaria el valor máximo de este índice alcanza 0,5. Conforme vaya disminuyendo el índice, aumenta la relevancia de la característica. Como contrapartida de este método, durante los cálculos no se eliminan datos redundantes.

2.6.5 Ganancia de la Información.

Esta técnica es una medida de la dependencia entre la característica y una clase. Es una de las técnicas más populares y simples, siendo calculada para una característica X y la clase Y de la siguiente manera:

$$InfoGain(X, Y) = H(X) - H(X|Y) \quad (2.44)$$

Donde H es la entropía:

- $H(X)$ – entropía de X
- $H(X|Y)$ – entropía de X después de haber observado Y

$$H(X) = - \sum_i P(x_i) \log_2(P(x_i)) \quad (2.45)$$

$$H(X|Y) = - \sum_j P(y_j) \sum_i P(x_i|y_i) \log_2(P(x_i|y_i)) \quad (2.46)$$

El valor máximo de *Info Gain* es 1. La característica más relevante tiene el valor máximo de este índice. Igual que en el anterior caso, esta técnica tampoco elimina información redundante.

2.6.6 Método de Mínima Redundancia y Máxima Relevancia.

Durante esta técnica se seleccionan las características que estén lo más lejos posible entre sí. A la vez estas características tienen que estar altamente correlacionadas con la variable de clasificación.

El objetivo final es maximizar la dependencia entre la distribución conjunta de las características seleccionadas y la variable de clasificación.

La Redundancia Mínima para una variable discreta se calcula de la siguiente forma:

$$\min W_I, W_I = \frac{1}{|S|^2} \sum_{i,j \in S} I(i,j) \quad (2.47)$$

Igualmente, la Relevancia Máxima para una variable discreta si obtiene así:

$$\min V_I, V_I = \frac{1}{|S|^2} \sum_{i,j \in S} I(h,i) \quad (2.48)$$

En ambas expresiones se define lo siguiente:

S – conjunto de características

I(i,j) – información mutua entre las características i y j

h - clases objetivo

A su vez, el criterio de la dependencia máxima I(S,h), que proporciona la información mutua de las variables seleccionadas S_m y la variable (clase) objetivo se puede definir de esta forma:

$$I(S_m; h) = \int \int p(S_m, h) \log \frac{p(S_m, h)}{p(S_m)p(h)} dS_m dh \quad (2.49)$$

2.6.7 Selección de características basada en correlación (CFS)

Durante esta técnica se evalúa la validez de las características mediante el uso de la correlación basada en heurística:

$$Mérito_S = \frac{k\bar{r}_{cf}}{\sqrt{k+k(k-1)\bar{r}_{ff}}} \quad (2.50)$$

El Mérito_S es un mérito heurístico de un conjunto de k características S.

Por otro lado, se define la media de la correlación entre la característica y clase:

$$\overline{r_{cf}} = \sum_{f_i \in S} \frac{1}{k} \sum (f_i, c) \quad (2.51)$$

Esta variable indica con que grado de facilidad se puede predecir una clase con una característica dada. El valor promedio de la inter-correlación de características se representa mediante \bar{r}_{ff} e indica posible redundancia.

La búsqueda de las mejores características se realiza mediante la técnica “Best First”, que es un algoritmo iterativo. Las iteraciones se paran cuando los resultados no se mejoran durante cinco intentos consecutivos.

Capítulo 3: Técnicas de clasificación aplicadas a las señales ECGs

3.1 Introducción

Tras la presentación de distintas técnicas de clasificación de una señal temporal en el capítulo 2, se requiere especificar el uso de estas técnicas con los electrocardiogramas.

A la hora de clasificar una señal mediante los métodos descritos en el apartado 2.1 lo único que puede variar para distintas señales, es la forma de su delimitación o extracción de un área de interés.

En un ECG la información más relevante se contiene en las formas de pulsaciones, variando éstas en función de una anomalía cardíaca u otra. Teniendo en cuenta esta particularidad, se describe a continuación el proceso de clasificación de un electrocardiograma, explicando las técnicas disponibles, eligiendo la plataforma, describiendo la base de datos empleada y finalizando con un ejemplo práctico de clasificación.

3.2 Estado del arte

Una vez obtenidas las pulsaciones digitalizadas, que corresponden a áreas de interés, según la mayoría de estudios de investigación relacionados con la detección de anomalías cardíacas a partir de los ECGs, es necesario implementar cuatro pasos para poder identificar las mismas: reducción de ruido, extracción de características, selección de características y clasificación.

A pesar de la unanimidad en los pasos a seguir, la forma de abordarlos es distinta, presentando cada autor una solución diferente.

Para la clasificación en la mayoría de los trabajos de investigación se ha podido comprobado la superioridad de SVMs, siendo descrita esta técnica en el apartado 2.4, frente a otros métodos según [Meh 07], [Kar 10] y [Can 10].

Para la reducción del ruido se aconseja el uso de la transformada de Wavelet, según las referencias [Jal 09], [Son 05] y [Can 12]. No obstante, la reducción del ruido en algunos casos no se requiere implementar de forma separada. Cuando la extracción de características se realiza mediante la WT, el ruido se elimina directamente, sucediendo lo mismo para la extracción de características mediante cumulantes, eliminándose el ruido Gaussiano de forma automática.

Para la extracción de características la mayoría de los autores ([Yin 12], [Jal 09] y [Son 05]) recomiendan el uso de la transformada de Wavelet, presentada su base matemática en el apartado 2.5.2. Otra posible forma de extracción es mediante cumulantes, implementada por autores de [Kar 10] y [Gua 12] y desarrollada en el apartado 2.5.1.

En el siguiente punto, reducción de características, hubo una multitud de propuestas, siendo algunas de ellas el Análisis Discriminante Lineal [Son 05], Algoritmos Genéticos [Jal 09], Análisis de Componentes Principales [Yin 12] y Selección de Características basada en Correlación (CFS) en [Can 10].

Algunas técnicas de reducción de características, como es el caso del Análisis de Componentes Principales y Análisis de Componentes Independientes, se han descartado directamente en este proyecto al no proporcionar una selección de características basada en algún tipo de dependencia entre característica y clase.

Otras alternativas de reducción de características encontradas han sido las propuestas por la Universidad Estatal de Arizona. Las técnicas recomendadas, descritas en el apartado 2.6, son el Método de Kruskal Wallis, el Test Chi-Square, el Índice de Gini, la Ganancia de la Información y el Método de Mínima Redundancia y Máxima Relevancia.

3.3 Base de datos y señales

La Base de Datos de las ECGs que se ha elegido es la base de arritmias MIT [Web N5]. Esta base de datos, utilizada en muchos estudios biomédicos, se caracteriza por una amplia representación de las anomalías cardíacas y una facilidad para la obtención de datos de interés.

La base de arritmias MIT dispone de 48 registros ECG, realizados en dos canales, de 30 minutos de duración. Las señales vienen digitalizadas con 360 muestras por segundo (frecuencia de muestreo) en los registros de datos con los registros de anotaciones médicas correspondientes.

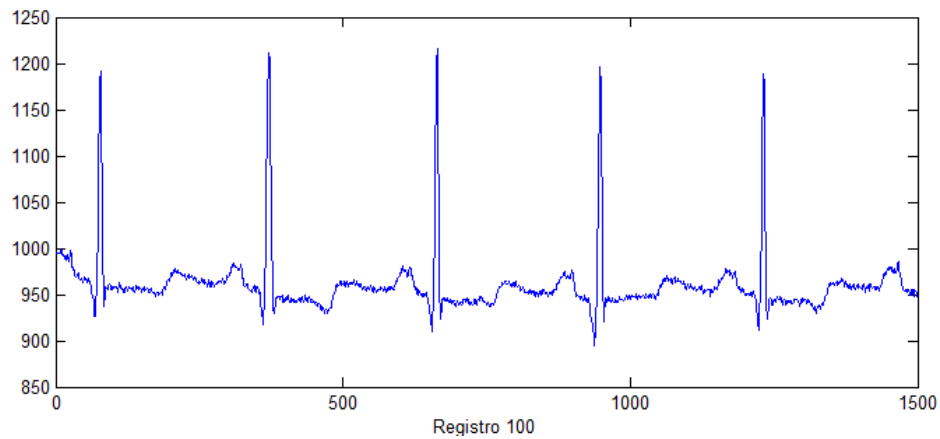


Fig. 3.1 Registro de datos N°100 de la base MIT

La obtención de pulsaciones cardíacas es esencial para poder continuar con las técnicas de procesamiento. Para esto se ha elegido un canal de la ECG (concretamente el N°1), se ha identificado el punto R que corresponde al punto máximo de una pulsación y se ha acotado un tramo de interés. Este tramo cubre 45 puntos equivalentes a 125 ms antes de R y 45 después. De esta manera, el ancho de pulso de las pulsaciones cardíacas, obtenidas de los registros de la base MIT, es igual a 250 ms.

Debido a que durante el procesamiento de datos a menudo se da prioridad a los datos de mayor valor numérico, tomándolos como más relevantes y confundiendo al algoritmo de clasificación, se ha acotado el valor máximo de las pulsaciones. La acotación se ha realizado mediante una traslación vertical, fijando el punto más alto R, en 1.250, y desplazando el resto de los puntos en una distancia $(1.250-R)$.

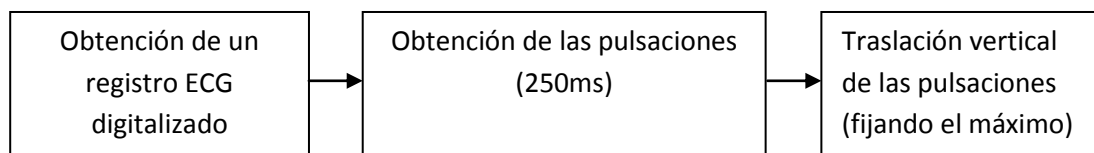


Fig. 3.2 Obtención de las pulsaciones

3.4 Plataforma

Las principales tareas a abordar durante los trabajos de clasificación son numerosos y complejos cálculos que van desde la obtención de las muestras de la base de datos, pasando por un preprocesamiento hasta la posterior obtención y reducción de las características, llegando a la clasificación final.

Debido a lo anterior, se precisan las herramientas de cálculo modernas y rápidas. Al haberse experimentado en la última década un gran auge de las aplicaciones basadas en las SVMs, se han desarrollado numerosos paquetes de esta técnica de clasificación en diversos lenguajes de programación.

Los paquetes de SVM más destacables bajo lenguaje de programación C son los que vienen a continuación:

- LIBSVM [9]
- SVMlight [26]
- SVMtorch [27]

Para la realización de este proyecto se ha pretendido encontrar el software que estuviera mejor actualizado, ampliamente documentado, siendo a la vez práctico, amigable y disponible para ejecutarlo en MATLAB. Debido a estos criterios, se ha elegido el paquete LIBSVM.

El toolbox LIBSVM ofrece la facilidad de entrenamiento y el cálculo de los parámetros de kernel. Una vez realizados los cálculos preliminares, se validan estos datos mediante el conjunto de muestras de test, verificando de esta forma la fiabilidad del modelo de clasificación obtenido.

Para los cálculos de los algoritmos de reducción de características, excepto LDA que se ha implementado directamente en MATLAB, se ha elegido el toolbox Feature Selection Algorithms de la Universidad Estatal de Arizona [Zhe 10], [Web N7].

El software elegido para el cálculo de cumulantes es el toolbox HOSA de la referencia [Web N2].

3.5 Ejemplo: clasificación de tres clases.

Una vez definidas las técnicas de clasificación aplicadas a las señales ECGs y elegida la plataforma para su implementación, se procederá a realizar una sencilla prueba de la misma. En esta prueba se van a clasificar tres tipos de pulsaciones cardíacas de la referencia [Gua 12]. Se estudiarán dos casos de extracción y selección de características: HOS con LDA (caso 1) y WT Db4 con Chi-Square (caso 2).

Las tres pulsaciones, tomadas de la base MIT [Web N5], corresponden a una pulsación normal N, y dos pulsaciones anómalas, R con bloqueo del ramal circulatorio derecho y L con bloqueo del ramal circulatorio izquierdo. Cada grupo está representado por 300 pulsaciones (muestras) de su tipo.

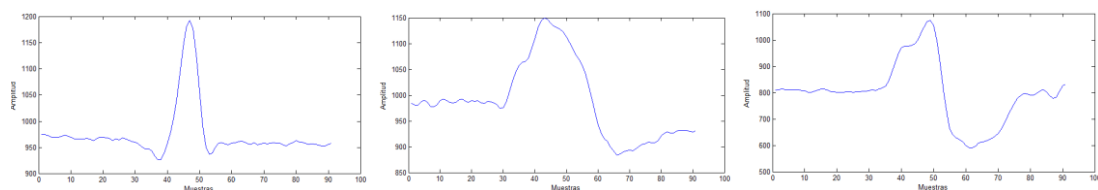


Fig. 3.3 Pulsaciones N, L y R respectivamente.

En el primer caso de la prueba que se va a implementar, se han elegido las técnicas de extracción y selección de características empleadas en la referencia [Gua 12]: cumulantes de la estadística de alto orden (descritos en el apartado 2.5.1) para la extracción y el ratio discriminante de Fisher del análisis discriminante lineal (desarrollado en el apartado 2.6.2) para la selección de características, realizando a continuación una clasificación mediante SVM.

De forma similar, el segundo caso corresponde a la extracción de características mediante la transformada de Fourier con base Daubechies 4, descrita en el apartado 2.5.2, siendo la técnica de selección Chi-Square, presentada en el apartado 2.6.3. Una vez extraídas las características relevantes, se realiza la clasificación de las mismas mediante tres clasificadores independientes de SVM.

De las 300 muestras de cada grupo se van a emplear las primeras 150 muestras (llamado este conjunto el de entrenamiento o “training”) para obtener un modelo-clasificador mediante datos históricos y los 150 restantes (llamadas las muestras de test) para probarlo.

El objetivo de esta prueba es obtener tres clasificadores, sin unirlos en ningún bloque lógico, que sean capaces identificar cada muestra por separado, con porcentajes de precisión cercanos al 100%.

El clasificador elegido es de tipo “One Against All”, o sea “Uno Contra Todos”. Este clasificador es de tipo binario (trabaja con dos grupos de datos) y cuenta con la presencia de todas las muestras. Cada clasificador compara una clase de muestras con las demás. Al haber tres tipos de pulsaciones, N, L y R, es necesario emplear tres clasificadores. El primero se encarga de identificar la clase N, comparándola con las pulsaciones tipo R y L. El segundo tiene por objetivo identificar la clase R, comparando sus muestras con las de N y L. Y por último, el tercer clasificador trata de identificar la clase L, mediante una comparación con muestras N y R.

Para la obtención de cada clasificador se precisan datos históricos, en este caso las pulsaciones del conjunto de entrenamiento. La presencia de clase que se está buscando tiene que ser la misma que las demás clases.

De esta manera, el clasificador N va a disponer de 150 muestras N y 150 muestras de R y L conjuntamente (75 de L y 75 de R).

Igualmente, el clasificador R tendrá para las tareas de entrenamiento 150 muestras de R y 150 muestras del conjunto N y L (75 de N y 75 de R).

Por último, el clasificador L dispondrá para el entrenamiento 150 muestras de L, 75 de N y 75 de R.

Como ya se había descrito en los apartados 2.1 y 3.2, es necesario primero extraer las características, luego reducirlas y por último clasificarlas (obteniendo un modelo mediante conjuntos de entrenamiento y después probándolo con datos de test).

3.5.1 Caso HOS y LDA

Antes de empezar con las tareas de clasificación es necesario primero procesar las pulsaciones de entrenamiento de cada clasificador mediante técnicas de extracción y reducción de características.

Para este fin, según lo descrito en el apartado 2.5.1, en primer lugar se ha procedido a extraer las características de cada pulsación empleando la estadística de alto orden (se pueden consultar las instrucciones para implementar estos cálculos en el Anexo B). El resultado obtenido consiste en dos cumulantes, uno de segundo y otro de cuarto orden. Al contar cada pulsación con 91 datos (correspondientes a 250 ms de ancho de pulsación), se han obtenido dos cumulantes de 91 datos cada uno.

Posteriormente, empleando la técnica del apartado 2.6.1 y las instrucciones del Anexo B, se ha seleccionado una característica para cada cumulante, siendo ésta una columna del cumulante que permite identificar correctamente la clase de interés, comparándola con otras clases. La característica más relevante corresponde a la columna del cumulante que tiene mayor valor del ratio discriminante de Fisher. Al haber dos cumulantes, se han obtenido dos características discriminantes para emplearlas en los clasificadores en vez de las pulsaciones originales.

Durante el cálculo del ratio discriminante de Fisher se han empleado todas las muestras (tanto las de entrenamiento como las de clasificación) para tener una visión más amplia de los datos de cumulantes que mejor identifican cada clase. No obstante, se pueden emplear sólo las muestras de entrenamiento porque los resultados varían muy poco.

En la siguiente tabla se representan las características relevantes halladas mediante FDR:

Tabla 3.1. Características representativas de los cumulantes

	NLR	LNR	RNL
Cumulante 2	23/69	5/87	30/62
Cumulante 4	61	5	52

Al observar la Tabla 3.1, se puede ver que en algunos casos se pueden utilizar de forma indistinta dos características (en este caso columnas) de un cumulante.

En las siguientes gráficas (3.4, 3.5 y 3.6) se representan como se separa la clase N (en verde) de las clases R y L, empleando la columna 23 del Cumulante 2 y la columna 61 del Cumulante 4.

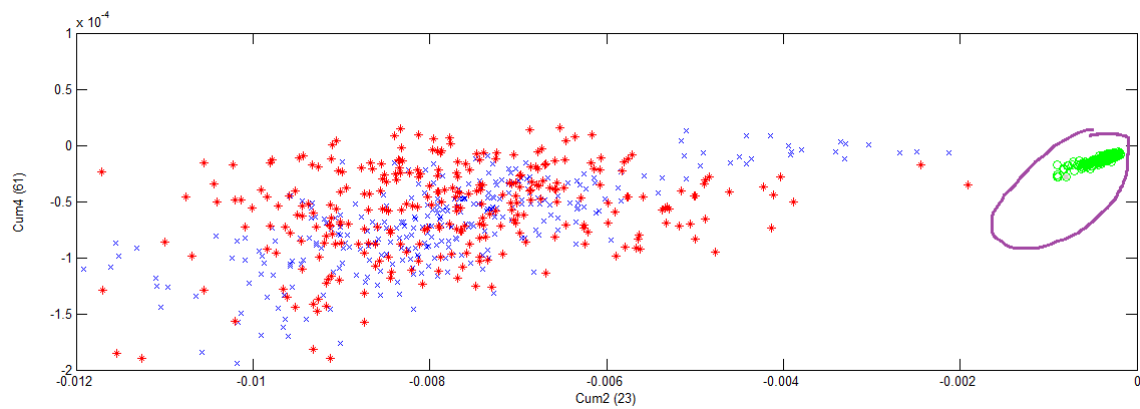


Fig.3.4 Separación de clase N mediante características relevantes de cumulantes

En las gráficas siguientes se representa la separación de la clase L (Fig. 3.5) y de la clase R (Fig.3.6) empleando características relevantes.

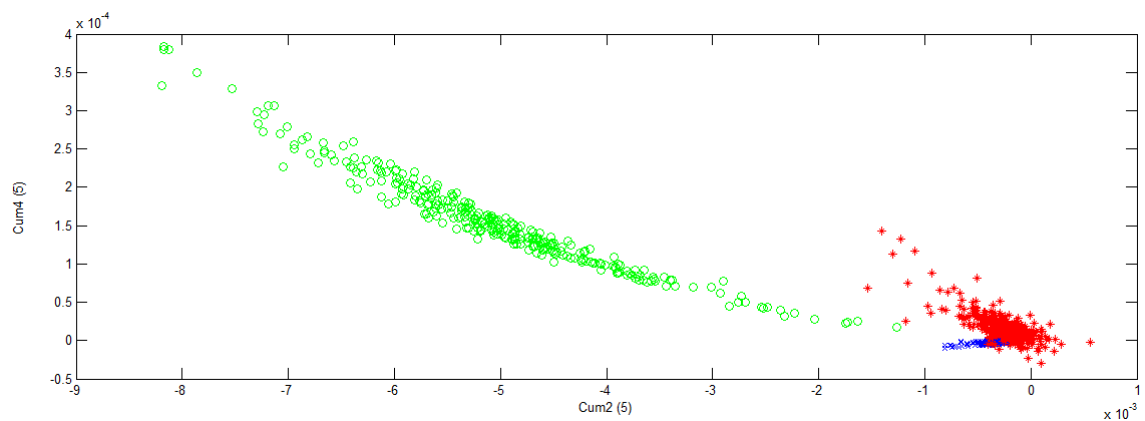


Fig.3.5 Separación de clase L (en verde) de las demás clases.

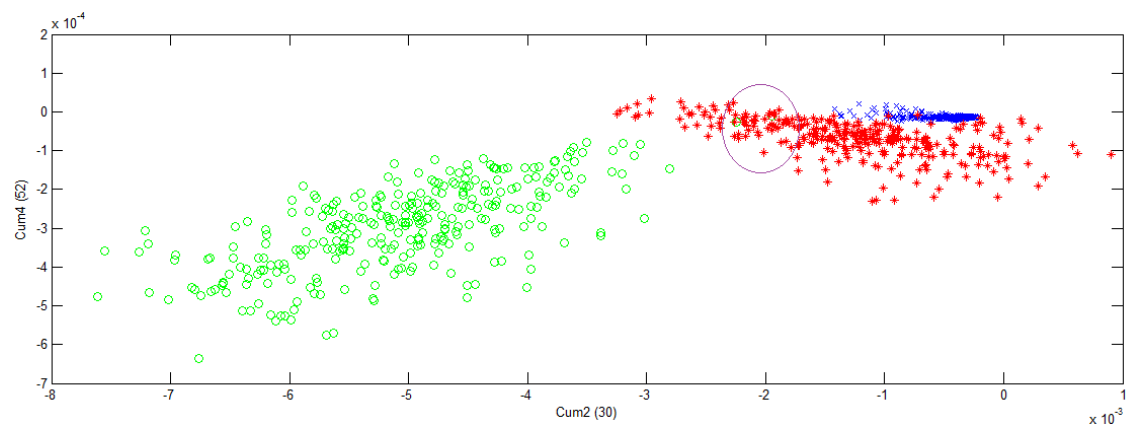


Fig. 3.6 Separación de clase R (en verde) de otras clases.

Para ver como el uso de una columna de un cumulante en concreto afecta a la separación de clase de interés de otras clases se va a examinar la Fig. 3.6. En la parte derecha de esta grafica hay dos muestras R entre las muestras L (el área está seleccionada con un círculo). Estas muestras no se han separado correctamente debido a una mala discriminación al emplear la característica Nº 30 del cumulante 2.

Las muestras mal discriminadas son las muestras número 274 y número 275 (representadas en la Fig. 3.7), con valores $-0,0022$ y $-0,0019$. Estos valores exceden bastante del rango habitual del cumulante 2 de la señal R, aproximándose a su vez a los valores de la señal L.

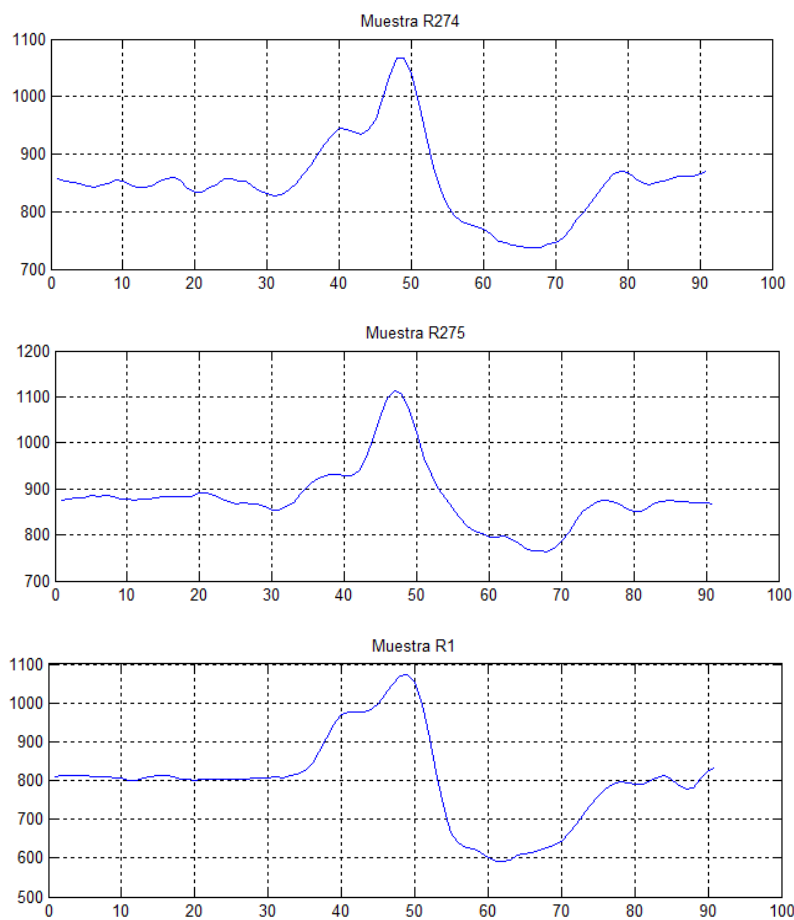


Fig. 3.7 Muestras de la señal R: atípicas 274, 275 y la habitual 1.

Utilizando otras características de los cumulantes para la separación de clase R, se puede ver como esto afecta a la discriminación (Fig. 3.9 para la característica 31 en vez de 30 y Fig. 3.8 para la característica 29 en vez de 30):

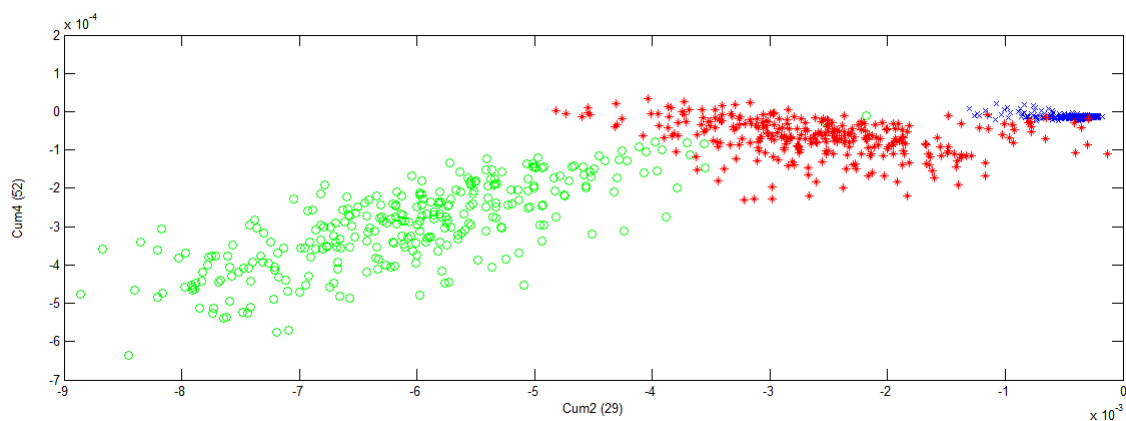


Fig. 3.8 Característica 29 del cumulante 2, clase R.

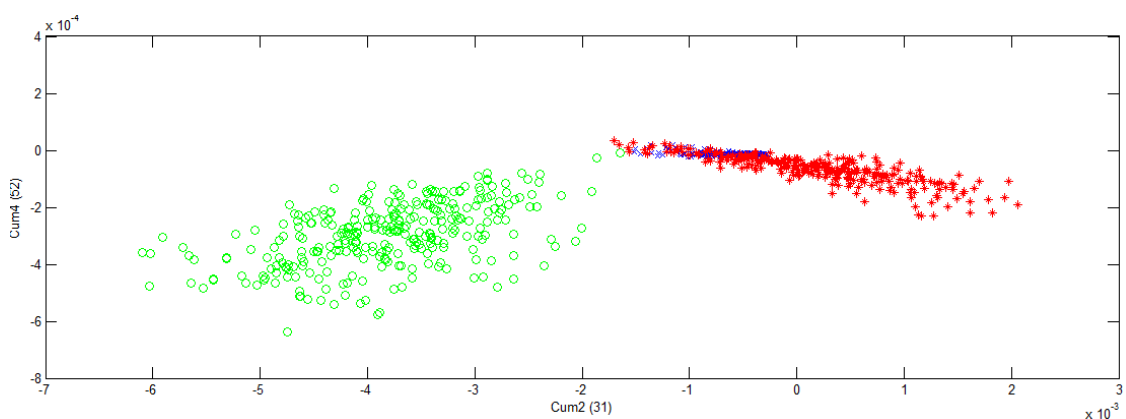


Fig. 3.9 Características 31 del cumulante 3, clase R.

Según las gráficas anteriores (Fig. 3.8 y 3.9), la característica 31 del cumulante 2 es mejor que la 30, ya que los dos grupos se separan por completo. Por otro lado, con la característica 29 los grupos se vuelven a mezclar.

La tabla definitiva de características de cumulantes (con la 31 en vez de la 30) queda así:

Tabla 3.2. Elementos representativos de los cumulantes.

	Clase N	Clase L	Clase R
Cumulante 2	23	5	31
Cumulante 4	61	5	52

Para obtener los modelos de clasificación es necesario emplear las matrices de entrenamiento. El conjunto de entrenamiento de clase N se representa mediante una matriz de 300 filas y dos columnas. Las primeras 150 filas corresponden a la columna 23 de cumulante 2 y a la columna 61 de cumulante, calculadas a partir de las pulsaciones de entrenamiento de N. Las siguientes 75 filas corresponden a la columna 23 de cumulante 2 y a la columna 61 de cumulante 4 calculadas a partir de las

pulsaciones de entrenamiento de L. Las 75 filas restantes se obtienen de forma análoga a partir de las pulsaciones de entrenamiento de R.

El mismo proceso se repite para la obtención de la matriz de entrenamiento de clase L, calculando sus primeras 150 filas a partir de las muestras L, siguiendo con 75 filas de N y terminando con 75 filas de R.

De forma análoga, la matriz de entrenamiento de R se obtiene a partir de 150 filas de R, continuando con 75 de N y finalizando con 75 de L.

Una vez obtenidas las matrices de entrenamiento de clase N, L y R, éstas se normalizan entre 0 y 1, siguiendo las instrucciones del Anexo A.

Además de las matrices durante la obtención del modelo de clasificación, se requiere especificar qué kernel se va a emplear e indicar sus parámetros. Los kernels empleados en este apartado son el kernel RBF y el kernel Lineal. Los parámetros de kernel se calculan empleando las matrices de entrenamiento y siguiendo las instrucciones del Anexo A.

Los parámetros de kernel obtenidos vienen a continuación en la Tabla 3.3.

Tabla 3.3. Parámetros de kernel.

	Clase N			Clase L			Clase R		
	C	G	%	C	G	%	C	g	%
RBF	32	0.5	100	32	2	100	32	0.0078125	100
Lineal	8	2	100	32	2	100	0.125	2	100

Después, utilizando los parámetros de kernel calculados y las matrices de entrenamiento, se obtienen modelos de clasificación (Anexo B). Cabe destacar los elevados aciertos de clasificación obtenidos para las matrices de entrenamiento que han sido del 100%.

Una vez obtenidos los modelos de clasificación para cada clase, se hace la prueba de los mismos utilizando las muestras de test.

Para esta prueba se han construido las matrices de test de la misma manera que las de entrenamiento, con la única deferencia que se han empleado las pulsaciones de test en vez de las de entrenamiento (150 últimas pulsaciones de N, L y R). En la tabla que sigue se muestran los resultados de clasificación para estas matrices de test.

Tabla 3.4. Resultados de Test.

	RBF		Lineal	
	Precisión %	Nº Vectores soporte	Precisión %	Nº Vectores soporte
Clase N	100	6	100	12
Clase L	100	8	100	10
Clase R	99,3	60	98,7	132

Las precisiones obtenidas miden la capacidad de un modelo de clasificación de reconocer si las muestras pertenecen a su clase o a otra distinta (sin especificar a qué clase en concreto). Una de las salidas que proporciona el modelo de clasificación es la identificación de clase {1, -1} que se llama etiqueta. Si la etiqueta toma valor 1, el modelo detecta que la muestra de test pertenece a su clase. De forma contraria, si la etiqueta toma valor -1, la muestra no pertenece a su clase, correspondiendo a una de las otras dos clases de la oposición.

La única precaución que se debe tener en cuenta, es a la hora de normalizar la muestra entre 0 y 1. Para esto, es preciso normalizar esta muestra dentro de la matriz de datos de test que se usó para la obtención del modelo de clasificación. Una vez normalizada la muestra dentro del conjunto inicial (siguiendo instrucciones del Anexo A), ya se puede emplear para la clasificación.

Para un caso de tres modelos es necesario normalizar la muestra tres veces (cada vez dentro de la matriz de test del modelo correspondiente).

Si se pretende clasificar más de una muestra, se normalizan una por una dentro de las matrices de test siguiendo el siguiente proceso: se añaden los datos de la muestra a continuación de la matriz de test, después se normaliza la matriz resultante entre 0 y 1, y, finalmente se extrae de la matriz su última fila que contiene la muestra normalizada.

Si no se hace así, los resultados de clasificación son pésimos porque las SVMs tratan de ubicar la muestra nueva entre los datos de test mediante vectores de soporte (que también provienen de los datos de test).

Viendo los resultados de clasificación de las matrices de test durante la prueba de los modelos, se puede comprobar en la tabla 3.4 que dichos modelos han proporcionado unas precisiones muy elevadas, cercanas al 100%.

Estos modelos sirven para poder identificar cualquier pulsación de clase N, L ó R, sin que sea necesario emplear una matriz de test compuesta por tres tipos de pulsaciones. Basta con calcular las características relevantes de cualquier pulsación de forma solitaria, empleando luego estas características en los modelos de clasificación

para identificar la muestra. En un caso ideal, para los tres modelos con aciertos del 100%, al suministrar dos características relevantes de una pulsación N a la vez a los tres modelos proporcionan las siguientes salidas identificativas: 1 para el modelo N, -1 para el modelo L y -1 para el modelo R.

No obstante, en la práctica los modelos no son perfectos y en algunos casos las respuestas son erróneas, siendo posible una situación con respuestas obtenidas para una pulsación N igual a 1,-1, 1 en vez de 1,-1,-1. Una posible solución a este caso se expone en el siguiente capítulo mediante la comparación de las probabilidades (se obtiene como una salida adicional del modelo de clasificación) de que cada respuesta sea cierta.

Según algunos autores [Gua 12] existe relación inversa entre el número de vectores de soporte empleados para definir el hiperplano de separación y la precisión obtenida durante la clasificación. Como se puede ver en la tabla 3.4, para las muestras N y L, con pocos vectores de soporte, la precisión ha sido del 100%. De forma contraria, para las muestras R, con mayor número de vectores de soporte, la precisión se ha visto reducida hasta 99% y 98% para el kernel RBF y lineal respectivamente.

En la siguiente prueba se procederán a simular la situación más desfavorable para cada modelo de clasificación, proporcionándoles 450 datos de las dos clases de oposición. Estos nuevos conjuntos de test se han obtenido de las muestras que no han participado en la construcción de los modelos. Para el modelo N se han empleado las últimas 225 muestras L y 225 muestras R, para el modelo L el conjunto de test se compone de las últimas 225 muestras de N y 225 muestras R, obteniéndose el conjunto para el modelo R de forma análoga.

Los resultados obtenidos han salido igual de satisfactorios que en el caso de la tabla 3.4.

Tabla 3.5. Test: 450 datos "-1" en cada clasificador.

	RBF		Lineal	
	Precisión %	Nº Vectores sop.	Precisión %	Nº Vectores soporte
Clase N	100	6	99,8	12
Clase L	99,8	8	100	10
Clase R	100	60	100	132

3.5.2 Caso WT Db4 y Chi-Square

Para este caso se emplearán las mismas matrices de entrenamiento y de test que se habían empleado para HOS y LDA. Así mismo, repitiendo los pasos descritos en los anteriores apartados, se fijan los parámetros de kernel para los tres modelos de clasificación (Tabla 3.6).

Tabla 3.6 Parámetros de kernel

	Clase N		Clase L		Clase R	
	C	G	C	G	C	g
RBF	32	0.5	32	8	32	0.5
Lineal	8	2	8	2	8	2

A continuación se presentan los resultados obtenidos durante la clasificación del conjunto test de 300 muestras, descrito en el anterior ejemplo. Se puede ver poca variación de resultados entre HOS y LDA frente a WT Db4 y Chi-Square. También se puede ver que el peor resultado corresponde al modelo con mayor número de vectores de soporte (15 para el kernel lineal).

Tabla 3.7 Resultados obtenidos para 300 muestras.

	RBF		Lineal	
	Precisión %	Nº Vectores sop.	Precisión %	Nº Vectores soporte
Clase N	100	13	100	15
Clase L	100	13	99,3	15
Clase R	100	5	100	5

Lo mismo sucede durante la clasificación del grupo de test de 450 muestras, habiendo una ligera diferencia en precisión al favor de HOS con LDA.

Tabla 3.8 Resultados de clasificación para 450 muestras.

	RBF Precisión %	Lineal Precisión %
Clase N	100	100
Clase L	99,6	99,3
Clase R	100	100

3.6 Conclusiones

En este capítulo se han presentado las técnicas de clasificación aplicadas a las ECGs. También se ha seleccionado la plataforma software, compuesta por la herramienta LIBSVM para realizar tareas de clasificación, el toolbox HOSA para el cálculo de cumulantes y el toolbox de la Universidad Estatal de Arizona para implementar distintas técnicas de clasificación. A continuación, se ha podido comprobar el buen funcionamiento tanto de las técnicas de clasificación elegidas como del software, realizado una prueba de clasificación de tres clases para dos casos de extracción y selección de características. El porcentaje elevado de aciertos se debe a una correcta normalización de datos, buena selección de los parámetros de kernel y a las técnicas de obtención y reducción de características elegidas.

Capítulo 4: Identificación de pulsaciones cardíacas usando clasificación SVM

4.1 Introducción

En el capítulo 3 se ha podido comprobar la elevada eficiencia de los modelos de clasificación basados en las técnicas de extracción y reducción de características, siendo el porcentaje de aciertos cercano al 100%.

No obstante, tal como se ha descrito en el apartado 3.5, en la práctica los modelos de clasificación pueden cometer errores llegando a darse el caso de que una pulsación se reconoce por varios modelos.

Teniendo en cuenta que el sistema de clasificación final pretende identificar 8 diferentes tipos de pulsaciones, es muy probable que los modelos cometan errores identificando algunas pulsaciones.

Además, sabiendo antemano que varias pulsaciones (en este proyecto dos de ocho) proceden de la fusión de otras dos pulsaciones que también participan en la clasificación, se espera una tasa de errores importante.

Una posible solución a estos errores ha sido propuesta en el siguiente apartado, basándose en las probabilidades de los modelos de clasificación. El sistema propuesto se usará en el proceso de clasificación empleando diferentes combinaciones de técnicas de extracción y reducción de características.

Finalmente, basándose en los aciertos, se elegirá la mejor combinación de las técnicas de extracción y reducción, exponiéndola detalladamente.

4.2 Sistema de identificación de pulsaciones cardíacas propuesto

4.2.1 Ejemplos de tipos de arritmias

En este apartado se van a exponer ejemplos de arritmias, entendiendo por éstas las anomalías cardíacas, que conllevan cambios en las formas de la pulsación. Tras haber estudiado los registros de la base MIT [Web N5], se han seleccionado 8 tipos de pulsaciones por su amplia representación y cantidad suficiente de muestras. De estas 8 pulsaciones, 1 corresponde a una pulsación normal y otras siete a pulsaciones anómalas.

Las gráficas de estas pulsaciones, donde aparecen todas las muestras seleccionadas vienen en la Fig. 4.1.

Las arritmias cardíacas se pueden originar por múltiples causas, siendo una de ellas el fallo durante las propagaciones de señales eléctricas descrito en [Web N8]. Un ejemplo de estos fallos es el bloqueo del ramal izquierdo o derecho de His, afecta a las contracciones de las cavidades internas de corazón, llamadas ventrículos. Esto hace que uno de los ventrículos se contraiga más lentamente, originando una pulsación de tipo L (con bloqueo del ramal circulatorio izquierdo) o de tipo R (con bloqueo del ramal circulatorio derecho).

Otro ejemplo de arritmia es la contracción ventricular prematura (V), que se produce cuando los ventrículos se contraen antes de lo debido.

De forma parecida, se da el caso de una arritmia de pulsación atrial prematura (A) [Web N9] que suele aparecer espontáneamente entre las pulsaciones normales. En esta arritmia la onda P aparece antes de lo debido.

Otro ejemplo de anomalías son las pulsaciones estimuladas, P (“paced beat”), que se obtienen por mediación de un marcapasos (“pacemaker”). Estos impulsos, producidos por un marcapasos, aparecen justamente antes del complejo QRS.

Otras dos pulsaciones anómalas, F y f, se producen cuando se unen dos pulsaciones en una. En el caso de F se produce una unión entre una pulsación ventricular y una normal. En el caso de f, la unión es producida entre una pulsación normal y una estimulada.

Todas estas arritmias seleccionadas vienen en la Tabla 4.1. En la misma se detallan los registros utilizados para la obtención de muestras y el número de éstas. De todos los tipos de pulsaciones se han seleccionado las de la Tabla 4.1 por su amplia representación y cantidad suficiente de muestras

Tabla 4.1. Tipos de pulsaciones seleccionadas.

Símbolo pulso	Significado	Registro	Nº Muestras
. o N	Pulsación normal	100	2.000
L	Con bloqueo del ramal circulatorio izquierdo	109	2.000
R	Con bloqueo del ramal circulatorio derecho	118	2.000
A	Pulsación atrial prematura	232	1.300
V	Contracción ventricular prematura	208	990
P	Pulsación estimulada (<i>paced</i>)	102	2.000
F	Fusión de pulsación ventricular y normal	208; 213	730
F	Fusión de pulsación normal y estimulada (<i>paced</i>)	104	660

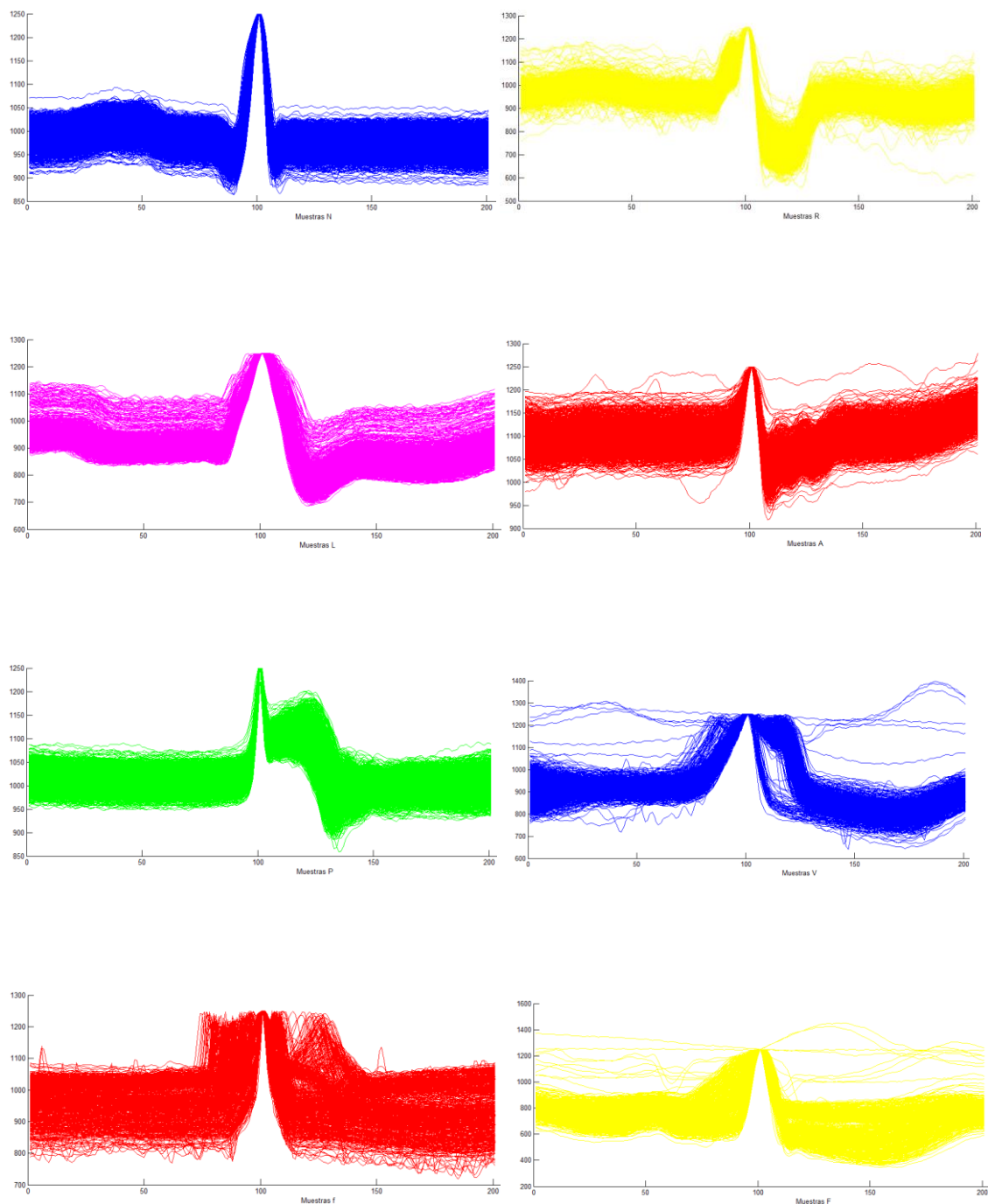


Fig. 4.1. Las muestras N, R, L, A, P, V, f y F (de izquierda a derecha y de arriba abajo).

4.2.2 Obtención de modelos de clasificadores binarios

Para la clasificación de las pulsaciones cardíacas, igual que en el Capítulo 3, se requieren modelos de clasificación binarios. Estos modelos se llaman binarios porque se trata de comparar dos grupos de datos, uno corresponde a la clase de interés y el otro a todas las demás clases.

Al haber ocho tipos de pulsaciones es necesario emplear ocho clasificadores OAA, correspondiendo un clasificador a cada clase.

Para la obtención de este clasificador son necesarias dos partes esenciales: las matrices de entrenamiento y los parámetros de kernel.

Las matrices de entrenamiento se obtienen de forma análoga que en el apartado 3.5, pero para más clases y más muestras. Por ejemplo, la matriz de entrenamiento de la clase N se compone de 2.100 filas y 6 columnas. Las primeras 1.050 filas corresponden a las características relevantes de las primeras 1.050 muestras de N. Las otras 1.050 filas se han repartido de forma equitativa entre las demás muestras: 150 primeras muestras de L, 150 primeras muestras de R, 150 primeras muestras de P, 150 de A, 150 de V, 150 de F y finalmente, 150 de f. Las columnas de esta matriz corresponden a seis características relevantes. Se ha optado por aumentar el número de características al haber más clases en la clasificación.

De la misma manera se han obtenido las matrices de entrenamiento de L, R, P y A.

Al disponer de menos muestras para las pulsaciones V, F y f, se ha optado por unas matrices de entrenamiento menores, siendo éstas de 1.120 filas y 6 columnas. De igual manera que antes, estas matrices se componen de 560 características relevantes de clase de interés, repartiendo las otras 560 muestras entre las demás clases. Por ejemplo, para la matriz de entrenamiento f se han tomado las siguientes características relevantes: las que corresponden a las primeras 560 muestras de f, añadiendo a continuación 80 filas correspondientes a las primeras muestras de N, siguiendo con las primeras 80 muestras de L, 80 de R, 80 de P, 80 de A, 80 de V y 80 de F.

De forma análoga, se han construido las matrices de entrenamiento de V y F.

Una vez obtenidas las matrices de entrenamiento, se puede proceder a calcular los parámetros de kernel siguiendo las instrucciones del Anexo A. El kernel que se ha elegido es RBF, descartando el kernel lineal por su forma de superficie de separación característica que en determinadas ocasiones no consigue separar bien las clases.

En el siguiente apartado se describe el clasificador de pulsaciones completo, que une los clasificadores binarios obtenidos en este apartado en un bloque lógico.

4.2.3 Clasificador de pulsaciones basado en SVM

Como ya se había expuesto en 3.5 y 4.1, en algunos casos varios clasificadores pueden reconocer a la vez una clase. Teniendo en cuenta que en este proyecto se trata de clasificar un amplio grupo de pulsaciones, es muy probable que se dé este caso.

Para solucionar esto se ha propuesto el sistema de clasificación representado en la Fig. 4.2. En este sistema los ocho clasificadores binarios del apartado anterior

están unidos en un bloque lógico. La entrada corresponde a las características relevantes de una muestra obtenida y normalizada de forma descrita en apartado 3.5.

Las características relevantes de la muestra de interés se clasifican a la vez mediante 8 clasificadores binarios. Cada clasificador tiene que responder si la muestra es de su clase o no. Aparte de la respuesta, se obtiene la probabilidad de la pertenencia a cada clase.

Una vez obtenidas las respuestas de los clasificadores, se comprueba cuantas respuestas positivas hay (cuando una clase se reconoce la respuesta es 1, y cuando no, -1)). En este proyecto sólo se han dado dos casos: habiendo una o varias respuestas positivas.

Cuando sólo hay una respuesta, la clase de la muestra es la del clasificador que la haya reconocido. Lo contrario sucede cuando se obtienen varias respuestas, siendo necesario elegir sólo una. El criterio de la elección condiciona los porcentajes de aciertos finales. Para este modelo se ha propuesto el criterio de elección basado en probabilidad.

El funcionamiento interno de un clasificador se reduce a lo siguiente: teniendo un espacio dividido en dos partes mediante un hiperplano y el grupo de datos de entrenamiento, se pretende ubicar la muestra de interés en uno de estos lados (si se queda en un lado, por ejemplo derecho, la muestra corresponde a la clase del clasificador; y, si se queda en el lado izquierdo, la muestra no corresponde a la clase del clasificador). Una vez ubicada la muestra en un lado (lo que equivale a determinar si es de una clase o no), ésta se queda a cierta distancia de la línea que divide el espacio en dos lados. En función de la lejanía o cercanía de la muestra a esta línea se puede deducir la probabilidad de que la respuesta sea cierta. Cada modelo de clasificación puede proporcionar esta probabilidad si así se especifica en sus parámetros [Chi 10].

De esta manera, comparando las probabilidades de varias respuestas, se puede escoger la que corresponda al mayor valor. No obstante, aunque este criterio en general funciona bien, en algunos casos la elección es errónea.

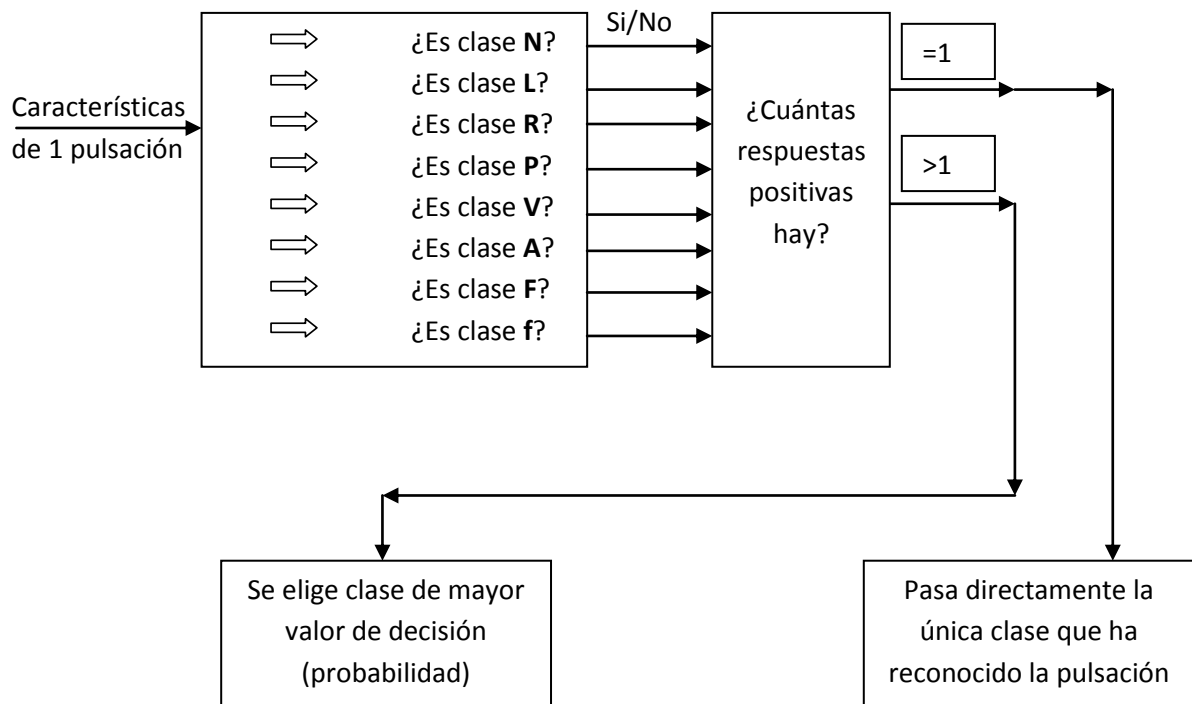


Fig. 4.2 Clasificador de las pulsaciones en bloque lógico.

4.2.4 Comparación de resultados obtenidos aplicando diferentes técnicas de extracción y reducción de características.

Una vez obtenidas las matrices de entrenamiento, definidos los clasificadores binarios que se van a emplear y propuesto un sistema de clasificación, se hace un estudio comparativo de las diferentes técnicas de extracción y reducción de características.

Las técnicas de extracción de características han sido tres: cumulantes de la estadística de alto orden (HOS), la transformada de Wavelet con la función base Daubechies 4 (Db4) y la transformada de Wavelet con la base Daubechies 6 (Db6).

Cada una de estas técnicas de extracción se ha probado con las siguientes siete técnicas de reducción de características:

- LDA: Análisis Discriminante Lineal
- χ^2 : Test Chi-Square
- Info Gain: Ganancia de la Información
- mRmR: Método de Mínima Redundancia y Máxima Relevancia
- CFS: Selección de características basada en correlación
- Kruskal Wallis: Método de Kruskal Wallis
- Gini: Test de Gini

Los cálculos a seguir para cada caso (existiendo 21 combinaciones en total) son similares a los realizados en los apartados 3.5.1 y 3.5.2, para un ejemplo de 3 clases de pulsaciones. Los pasos seguidos para obtener los porcentajes de aciertos en este apartado para WT Db4 y Chi-Square son parecidos a los pasos del apartado 3.5.2, comprendiendo la elección de los parámetros de kernel, la obtención de características, la reducción de las mismas, la obtención de los modelos de clasificación y la verificación final de los modelos obtenidos. Aparte de esto, en el presente apartado se requiere trabajar con ocho clases de pulsaciones en vez de tres y construir un bloque lógico de clasificadores basado en la probabilidad de aciertos de los modelos.

Todo esto se requiere realizar también en veinte casos restantes, ejecutando para cada uno los pasos descritos arriba.

Los resultados finales de todas estas técnicas, correspondientes a la clasificación de las muestras que no han participado en la obtención de clasificadores (ultimas 950 muestras de N, L, R, P; ultimas 250 muestras de A; ultimas 430 muestras de V; ultimas 170 muestras de F y ultimas 100 de f) se relacionan a continuación:

Tabla 4.2 Resultados obtenidos aplicando diferentes técnicas de procesamiento.

	LDA	Chi²	Info Gain	mRmR	CFS	Kruska Wallis	Gini	<i>Acierto Medio</i>
HOS	99,1	97,3	99	97,9	98,9	97,7	97,9	98,3
Db4	93,6	94,6	94,7	96,4	97,7	82,9	93,8	93,4
Db6	95,1	95,5	95,8	94,1	89,7	62,4	97,1	90

Cada valor numérico de la tabla 4.2 corresponde a los aciertos medios (de ocho clases de pulsaciones) obtenidos para cada combinación de una técnica de extracción de características con una de selección. Las técnicas de extracción vienen representadas en la primera columna: HOS, Db4 y Db6. A su vez, las técnicas de selección corresponden a la primera fila de la tabla (de LDA a Gini incluido). En la última columna se recogen los aciertos medios para cada técnica de extracción empleada.

El mejor resultado se ha alcanzado utilizando la estadística de alto orden para la obtención de características, habiendo pocas diferencias entre los resultados de distintas técnicas de reducción. La gran eficiencia de HOS se debe a que los dos cumulantes obtenidos son independientes entre sí. En el caso de la transformada de Wavelet sólo se obtiene un grupo de características para cada pulsación, siendo su eficiencia menor para ambas bases (Db4 – Daubechies 4 y Db6 - Daubechies 6).

No obstante, comparando los resultados de la base Db4 con los de la DB6, se observa que la forma de onda de la primera base coincide más con las pulsaciones, proporcionando mayores aciertos.

4.2.5 El mejor clasificador encontrado: HOS y LDA

Según la Tabla 4.2, el mejor método obtenido se basa en la estadística de alto orden y el análisis discriminante lineal, descrito en la referencia [Gua 12].

Los pasos detallados durante la implementación de este método vienen a continuación.

Obtención de las pulsaciones

En primer lugar se han elegido los registros digitalizados (Tabla 4.1) de la base ECGs MIT con ocho pulsaciones que se quieren clasificar. Cada pulsación es de 91 puntos que corresponden a 45 puntos previos al máximo y 45 posteriores.

Teniendo en cuenta que la frecuencia de muestreo es $f=360\text{Hz}$, el ancho estudiado de cada pulsación sería igual a $2 \frac{45 \cdot 1000}{360} \text{ms}$, o sea 250ms .

Luego, las pulsaciones se han ordenando verticalmente, extrayendo a continuación la media (de la matriz compuesta por todas las muestras obtenida).

Cálculo de cumulantes

En el siguiente paso se han calculado los cumulantes de segundo y cuarto orden mediante la HOS, de acuerdo a 2.4.1. En las figuras que vienen a continuación (4.3, 4.4 y 4.5) se pueden ver los cumulantes para las pulsaciones N, R y L.

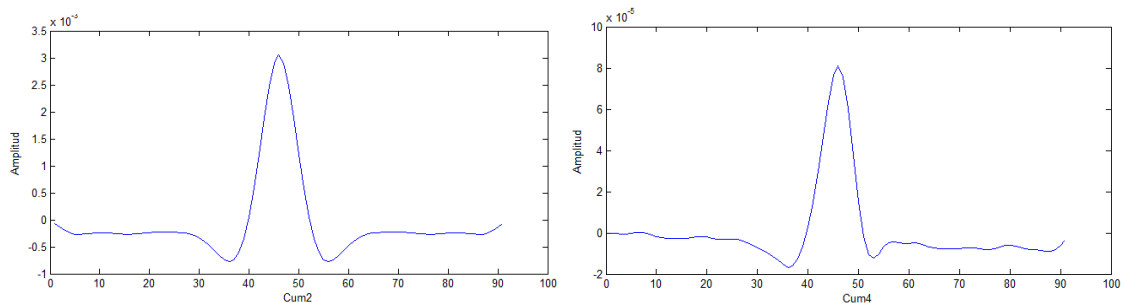


Fig.4.3 Pulsación normal: Cumulante 2 y 4 respectivamente

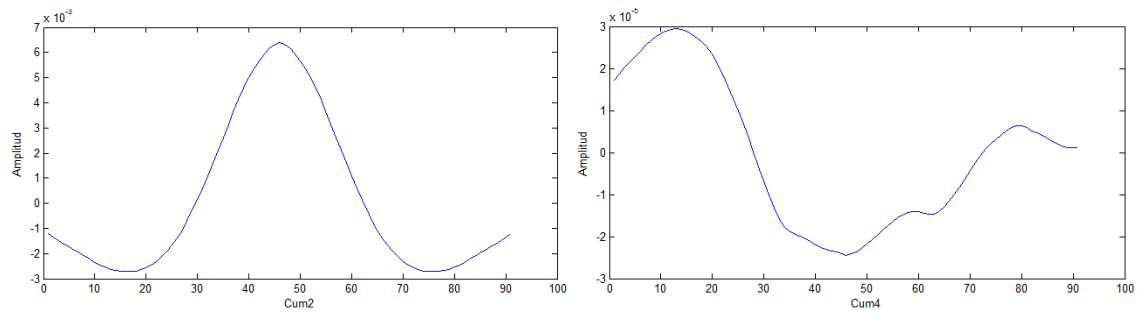


Fig.4.4 Cumulante 2 y 4 de la pulsación L.

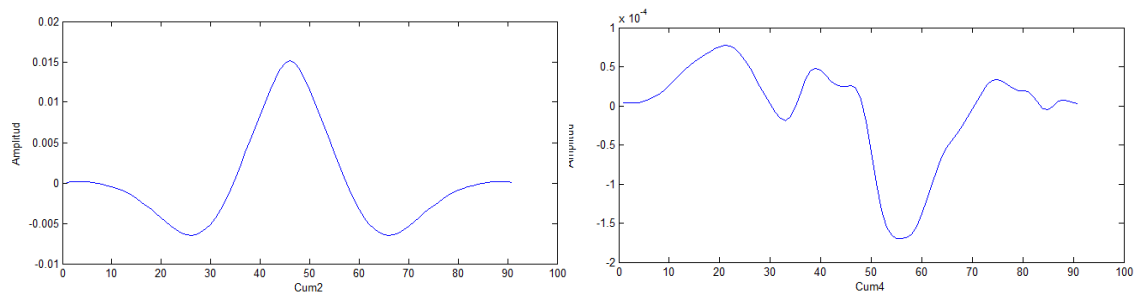


Fig.4.5 Cumulante 2 y 4 de la pulsación R.

De esta manera, para cada pulsación de 91 componentes, se han obtenido dos cumulantes que a su vez cuentan con 91 componentes.

Reducción de características

A continuación, se ha calculado el ratio discriminante de Fisher para cada Cumulante según (2.37) del apartado 2.5.1. Cada elemento de Cumulante 2 y 4 se ha valorado en función de su capacidad de separar las clases.

Con el fin de seleccionar las mejores características, se han elegido tres componentes de cada Cumulante con mayor ratio de Fisher. En la siguiente tabla aparecen los resultados correspondientes.

Tabla 4.3. 6 características relevantes para cada clase.

	Clase N	Clase L	Clase R	Clase P	Clase A	Clase V	Clase F	Clase f
Cumulante 2 (1)	37	21	29	25	46	1	46	23
Cumulante 2 (2)	55	71	63	67	45	91	45	69
Cumulante 2 (3)	38	22	30	26	47	2	47	24
Cumulante 4 (1)	22	67	37	66	56	91	67	54
Cumulante 4 (2)	21	66	36	65	57	90	68	53
Cumulante 4 (3)	20	68	38	64	58	89	66	55

Calculo de los parámetros de kernel y el entrenamiento

Seguidamente, se ha procedido a separar las pulsaciones en dos grupos: el de entrenamiento (training) y el de verificación (test).

Las matrices de entrenamiento se han formado para las pulsaciones N, L, R, P y A a partir de 1.050 muestras de interés y 1.050 de otros tipos de muestras. El tamaño resultante de cada matriz es de 2.100 filas (1.050 de datos de clase 1 y 1.050 de clase -1 formada por 7 tipos pulsaciones) y 6 columnas (que corresponden a las características relevantes de la Tabla 4.3).

De forma parecida se han formado las matrices para las pulsaciones V, F y f, que resultaron cada una de 1.120 filas (560 datos de clase 1 y 560 datos de clase -1) y 6 columnas.

El resto de las muestras se han quedado para la posterior verificación del modelo de clasificación obtenido.

Las matrices de entrenamiento se han normalizado entre 0 y 1 empleando (A.1) del Anexo A, calculando enseguida los parámetros de kernel RBF (siguiendo Anexo A).

Tabla 4.4. Parámetros de kernel para 6 características.

	Clase N	Clase L	Clase R	Clase P	Clase A	Clase V	Clase F	Clase f
C	128	32768	8192	32768	2048	128	32768	8192
g	8	8	0,5	8	2	8	8	8
Precisión %	99,9	99,1	99,7	99,3	99,9	99,1	98,4	96,9

Empleando los parámetros de kernel de la tabla 4.4, se han obtenido los modelos de clasificación para cada clase por separado.

Clasificación

Finalmente, se ha realizado la verificación final del modelo de clasificación obtenido, siendo sus resultados satisfactorios para todo tipos de pulsaciones (Tabla 4.4).

Tabla 4.5 Resultados HOS y LDA

	Muestras clasificadas	Acierto %
Clase N	950	99,3
Clase L	950	99,9
Clase R	950	99,5
Clase P	950	100
Clase A	250	100
Clase V	430	96,7
Clase F	170	92,9
Clase f	100	97
Acierto Medio	%.:	99,1

Cabe destacar que los aciertos obtenidos corresponden a las muestras de test que han sido identificadas correctamente. De todo el conjunto de muestras ha habido un grupo reducido que ha sido identificado de forma errónea, llamándose este error Falso Negativo (al no haberse reconocido las muestras de forma debida). Para este apartado el error Falso Negativo es menor de un 1% (0,9%).

4.3 Conclusiones

En este capítulo se ha propuesto un sistema de clasificación basado en probabilidad y en clasificadores binarios unidos en un bloque lógico. Se han estudiado varias alternativas de extracción y selección de características, probándolas mediante el sistema de clasificación.

Se ha podido comprobar que las técnicas de aprendizaje máquina empleadas han sido de gran utilidad para las tareas relacionadas con las búsquedas de patrones. A pesar de haber utilizado ocho tipos de pulsaciones con pocas diferencias visibles entre ellas, el modelo de clasificación obtenido ha sido capaz de identificar las muestras correctamente.

Finalmente, se ha expuesto detalladamente la mejor combinación encontrada, correspondiente a la estadística de alto orden y el análisis discriminante lineal, siendo su acierto medio del 99% (100% en el mejor y 93% en el peor caso).

Capítulo 5: Conclusiones y trabajos futuros

5.1 Conclusiones

En este proyecto se han estudiado las técnicas de clasificación, empleando las utilizadas en otros trabajos relacionados y las alternativas encontradas. También se ha hecho el estudio de las herramientas software disponible para implementar estas técnicas, siendo la más destacada la herramienta Libsvm.

Una vez descritas detalladamente los pasos a seguir durante la clasificación, se ha demostrado la alta eficacia de las técnicas mediante un ejemplo de tres clases. Los aciertos, cercanos al 100%, se deben también a la correcta normalización y a una buena selección de los parámetros de kernel.

Para abordar un caso de clasificación de 8 clases, se ha propuesto un sistema de clasificación basado en probabilidad y clasificadores binarios unidos en un bloque lógico. Este sistema ha demostrado un buen comportamiento con todas las combinaciones de las técnicas de extracción y selección probadas. La combinación que ha destacado por encima de todas, con un acierto medio del 99% (siendo el acierto del 93% para el peor y 100% para el mejor caso respectivamente), corresponde a la estadística de alto orden y el análisis discriminante lineal.

Finalmente, se ha expuesto detalladamente la mejor combinación, correspondiente a la estadística de alto orden y el análisis discriminante lineal, siendo su acierto medio del 99 %, acorde con los aciertos obtenidos en otros trabajos de investigación.

5.2 Trabajo futuro

Aunque el campo de aplicación de las técnicas de clasificación es inmenso, podrían ser de interés las siguientes posibles líneas de investigación:

- Probar la técnica de clasificación del 4.2.4 con otros tipos de señales.
- Implementar esta técnica en el sistema embebido de instrumentación de la referencia [Gua 12].

BIBLIOGRAFÍA Y REFERENCIAS

- [Bus 55] R. R. Bush, F. Mosteller *“Stochastic Models for Learning”*, John Wiley & Sons, 1955
- [Can 10] Can Ye, Miguel Tavares Coimbra and B.V.K. Vijaya Kumar, *“Arrhythmia Detection and Classification using Morphological and Dynamic Features of ECG Signals”*, 32nd Annual International Conference of the IEEE EMBS, Argentina, August 2010
- [Can 12] Can Ye, Miguel Tavares Coimbra and B.V.K. Vijaya Kumar, *“Heartbeat Classification Using Morphological and Dynamic Features of ECG Signals”*, IEEE Transactions on biomedical engineering, vol. 59, N. 10, october 2012
- [Chi 10] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin *“A practical Guide to Support Vector Classification”*, Department of Computer Science, National Taiwan University, 2010
- [Col 01] R. Collobert, S. Bengio & C. Williamson *“SVMtorch: Support Vector Machines for Large-Scale Regression Problems”*, Journal of Machine Learning Research, vol. 1, pp. 143–160, 2001
- [Est 50] W. K. Estes *“Toward a Statistical Theory of Learning”*, Psychological Review, vol. 57, No. 2, pp. 94-107, 1950
- [Fer 09] D.D. Ferreira, A.S. Cerqueira, C.A. Duque and M.V. Ribeiro *“HOS-based method for classification of power quality disturbances”*, The Institution of Engineering and Technology, Electronics Letters, vol. 45, no. 3, pp. 183-185, 2009
- [Ger 03] Carlos Gershenson *“Artificial Neural Networks for Beginners”*, Article, Computing Research Repository, 2003
- [Gua 11] G. A. Guarnizo Herreño, Marta Ruiz Llata *“Sistema de identificación de señales arrítmicas sobre ECG basado en HOS y SVM”*, Artículo, XXVI Edición de URSI, septiembre de 2011
- [Gua 12] G. A. Guarnizo Herreño *“Estudio de Máquinas de Vectores de soporte en sistemas embebidos para instrumentación”*, Trabajo fin de máster, UC3M, 2012
- [Heb 49] D. O. Hebb *“The organization of behavior: A neuropsychological approach”*, John Wiley & Sons, 1949
- [Jal 09] Jalal A. Nasiri , Mahmoud Naghibzadeh , H. Sadoghi Yazdi , Bahram Naghibzadeh *“ECG Arrhythmia Classification with Support Vector*

Machines and Genetic Algorithm", Third UKSim European Symposium on Computer Modeling and Simulation, 2009

- [Joa 99] T. Joachims ***"Making large-scale support vector machine learning practical"***, pages 169–184. Advances in kernel methods. MIT Press, 1999
- [Kar 10] S. Karpagachelvi ***"Classification of electrocardiogram signals with support vector machine and relevance vector machine"***, International Journal of Engineering Science and Technology vol. 2(11) pp. 6511-6520, 2010
- [Mah 05] S. Z. Mahmoodabadi, A. Ahmadian, M. D. Abolhasani ***"ECG feature extraction using Daubechies Wavelets"***, International Conference Visualization, Imaging and Image processing, Spain, 2005
- [Meh 07] S.S. Mehta and N.S. Lingayat ***"SVM for Cardiac Beat Detection in Single Lead Electrocardiogram"***, IAENG International Journal for Applied Mathematics, May 2007
- [Men 91] J.M. Mendel ***"Tutorial on Higher-Order-Statistics (Spectra) in Signal Processing and System Theory: Theoretical Results and Some Applications"***, IEEE, March 1991
- [Nir 13] H.S. Niranjana Murthy, M.Meenakshi ***"ECG Signal Denoising and Ischemic Event Feature Extraction using Daubechies Wavelets"***, International Journal of Computer Applications (0975 – 8887) Volume 67– No.2, April 2013
- [Ozl 10] N. Özlem Özcan, Fikret Gürgen ***"Fuzzy Support Vector Machines for ECG Arrhythmia Detection"***, International Conference on Pattern Recognition, 2010
- [Pan 10] S.C. Pandey, G.C. Nandi ***"Genetic Algorithm based Comparison of Different SVM"***, International Journal of Computer Applications (0975 – 8887), vol. 6, No.8, September 2010
- [Pon 97] M. Pontil, A. Verri ***"Properties of Support Vector Machines"***, Technical report, Massachusetts Institute of Technology Cambridge, 1997
- [Riv 09] E. Rivas Trujillo ***"Detección de averías en cambiadores de tomasen carga de transformadores basado en el patrón de vibraciones"***, Tesis doctoral UC3M, 2009
- [Ros 04] L. Rosasco, E.D. Vito ***"Are Loss Function All The Same?"***, Article, Neural Computation, vol 16, pp 1063-1076, 2004

- [Saf 04] M. Safari, M. T. Harandi and B.N. Araabi ***“A SVM-based method for face recognition using a Wavelet-PCA representation of faces”***, ICIP International Conference, vol. 2, pp. 853-856, October 2004
- [Son 05] Mi Hye Song, Jeon Lee, Sung Pil Cho, Kyoung Joung Lee, and Sun Kook Yoo ***“Support Vector Machine Based Arrhythmia Classification Using Reduced Features”*** International Journal of Control, Automation, and Systems, vol. 3, no. 4, pp. 571-579, December 2005
- [Swa 98] A. Swami ***“Higher-Order Spectral Analysis Toolbox”***, Mathworks, 1998
- [The 10] Sergio Theodoridis ***“Introduction to Pattern Recognition. A Matlab approach”***, Elsevier, 2010
- [Vap 74] V. Vapnik, A. Chervonenkis ***“Theory of Pattern Recognition”***, Nauka Moscow, 1974
- [Vap 79] V. Vapnic ***“Estimation of Dependencies Based on Empirical Data”***, Nauka Moscow, 1979
- [Yin 12] Thaweesak Yingthawornsuk ***“Classification of Cardiac Arrhythmia via SVM”***, 2012 2nd International Conference on Biomedical Engineering and Technology IPCBEE vol. 34, 2012
- [Zhe 10] Zheng Zhao ***“Advancing Feature Selection Research – ASU Feature Selection Repository”*** School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85287, 2010.
- [Web N1] Librería LIBSVM <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- [Web N2] HOSA toolbox
<http://www.mathworks.com/matlabcentral/fileexchange/3013>
- [Web N3] Python 3.3.1 <http://www.python.org/getit/>
- [Web N4] Gnuplot 4.6.0
<http://www.windows7download.com/win7-gnuplot-for-indows/download-draeklow.html>
- [Web N5] Base de datos ECG MIT
<http://www.physionet.org/physiobank/database/mitdb/>
- [Web N6] WFDB Tollbox for Matlab
<http://physionet.org/physiotools/matlab/wfdb-swig-matlab/>

[Web N7] Matlab toolbox Feature Selection
<http://featureselection.asu.edu/software.php>

[Web N8] Instituto de corazón de Texas <http://www.texasheartinstitute.org/>

[Web N9] Web educacional de ECG <http://www.elsitioecg.com/default.cfm>

ANEXO A: “Cálculos de los parámetros de kernel mediante Libsvm”

Durante la clasificación se han a utilizar dos tipos de kernel: RBF y Lineal. Sus parámetros se han a calcular con la función **grid.py** (incluida dentro del paquete de software Libsvm). Para poder ejecutar la función es necesario instalar **Python 3.3.1** [Web N3] y **Gnuplot** [Web N4].

Una vez instalados estos programas, se ha procedido a localizar la función svm-train.exe (incluida en el Libsvm) y gnuplot.exe. Luego, abriendo la ventana de comandos (Fig. 9) de Windows (Inicio->Todos los Programas->Accesorios->Símbolo de sistema), se retrocede mediante el comando **cd..** hasta el directorio C:\ y se calculan los parámetros de cada kernel.

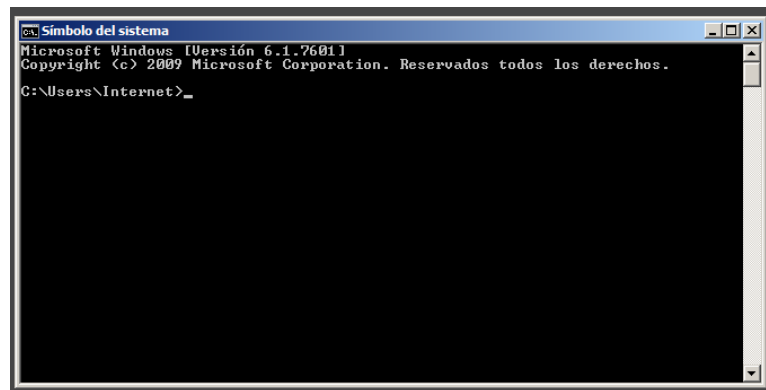


Fig. A.1 Ventana de comandos

En primer lugar, al abrir la ventana de comandos se indica el directorio de Python:

```
C:\> PATH C:\Python33;%PATH%
```

Luego, se procede a calcular los parámetros de kernel RBF:

```
C:\> python C:\Python33\grid.py -v 10 -svmtrain "C:\Python33\svm-train.exe" -  
gnuplot "C:\Program Files (x86)\gnuplot\bin\gnuplot.exe" -out  
"C:\Python33\dat.txt.out" C:\Python33\NLR.txt
```

En el código de arriba se ha especificado que se va a ejecutar un comando de Python y dónde se encuentra este comando. También se indicó que los parámetros de kernel se van a calcular mediante una *cross-validation* de diez ciclos (-v 10). Luego sigue en qué directorio están las funciones svm-train.exe y gnuplot.exe. Más abajo, se

escribe el nombre y el directorio donde se van a guardar los resultados y se especifica el archivo de texto en formato libsvm con las etiquetas y la matriz de datos.

Para los parámetros de kernel lineal según [16], se indica $-\log2g\ 1,1,1$ entre el comando `grid.py` y $-v\ 10$.

Debido a que los parámetros de kernel se van a calcular mediante Libsvm, es necesario preparar un archivo de texto con la matriz compuesta por los datos para el entrenamiento y las etiquetas.

Las etiquetas son las siguientes: 1 para los datos de una clase y -1 para los datos de otra clase, dentro del mismo grupo de datos.

Para poder utilizar la matriz de datos en cualquier función de Libsvm es necesario normalizarla entre 0 y 1. A raíz de que no existe una función específica para esto en Matlab, la normalización se realizará mediante este código:

```
M_score=M-repmat(min(M,[],1),size(M,1),1))*spdiags (1./max(M,[],1)-min(M,[],1))',0,  
size(M,2),size(M,2) (A.1)
```

M – es la matriz que se pretende normalizar

M_score – matriz normalizada

Una vez normalizada la matriz, es necesario ejecutar la función `sparse()` de Matlab, ya que las funciones para el cálculo de kernel, las de entrenamiento y las de clasificación de Libsvm sólo admiten datos de este tipo.

Seguidamente, se obtiene el archivo de texto en formato de Libsvm mediante la función `libsvmwrite()`. Habrá un archivo para cada grupo (por ejemplo, para tres clases los archivos serían NLR.txt, LNR.txt y RNL.txt).

ANEXO B: “Otros cálculos de interés realizados en Matlab “

Extrayendo las pulsaciones de la base MIT

La base MIT cuenta con 48 ECGs, donde cada grabación viene representada por tres archivos:

- Anotaciones de referencia: **100.atr**
- Señal : **100.dat**
- Encabezado: **100.he**a (100 es el N° de la ECG).

Para obtener las muestras se han descargado todos los archivos (xxx.atr, xxx.dat y xxx.he) al directorio de trabajo de Matlab. Luego, se instaló el WFBT toolbox de Matlab [Web N6]. Este toolbox ha dado problemas para funcionar desde el directorio habitual (a pesar de haberlo agregado al listado del Path), pero al cambiar al directorio donde estaba instalado (C:\Users\Internet\wfdb-swig-matlab\0.0.1) funcionó con normalidad.

Las funciones que se han usado para extraer las muestras son éstas:

rdann('100','atr','type','/') - devuelve en qué posiciones están las muestras del tipo especificado (en el ejemplo P Paced beat, '/') del registro 100.

rdsamp('100') – devuelve una matriz con datos de la ecg del canal 1 (columna 2) y canal 2 (columna 3).

Estas dos funciones vienen en forma de ejemplo en un fichero **demo_wfdb_tools.m** que ha sido de bastante utilidad a la hora de extraer las muestras.

Calculando los cumulantes

Para el cálculo de cumulantes se utiliza **toolbox HOSA** (High Order Spectral Analysis Tollbox) [Web N2]. La función que se emplea para obtener los autocumulantes 2 y 4 de cada pulsación es la siguiente:

cumest(y, norder, maxlag, sam_seg, overlap, flag, k1, k2)

donde:

y – es el vector o matriz de datos

norder – el orden del cumulante a calcular (2, 3 ó 4).

maxlag – de este valor depende el tamaño del vector de salida ($=2*\text{maxlag}+1$)

sam_seg – el número de muestras por segmento

overlap – el porcentaje de solapamiento entre segmentos

flag – puede tomar uno de estos valores: 'biased' o 'unbiased'

k1 y k2 – controlan que parte de 1D-segmento se calcula para el cumulante 3 ó 4.

Para el caso en estudio se utilizaron las siguientes configuraciones:

y: un vector con 91 muestras para cada pulsación

norder: 2 ó 4 para el cumulante correspondiente

maxlag: 45 (para obtener 91 datos para cada salida, igual al número de datos de la entrada)

Los demás parámetros quedaron con valores por defecto: $\text{overlap}=0$, $\text{flag}=\text{'biased'}$ y $k1=k2=0$.

Calculando FDR

El ratio discriminante de Fisher se ha programado mediante la implementación de formula (2.37) en Matlab, utilizando la función **mean()** para calcular las medias y **var()** para las varianzas.

Calculando la WT

Los comandos de Matlab que se han empleado para la obtención de los coeficientes de aproximación son los dos siguientes:

[C L]=wavedec(Matriz, k, 'db4');

Ap_coef=appcoef(C,L,'db4', k) k – nivel de descomposición (se ha empleado 5).

En la siguiente figura (Fig. A.2) se pueden observar los coeficientes de aproximación en el lado izquierdo y los de detalle en el lado derecho para cinco niveles de descomposición de una señal normal utilizando de base Db4.

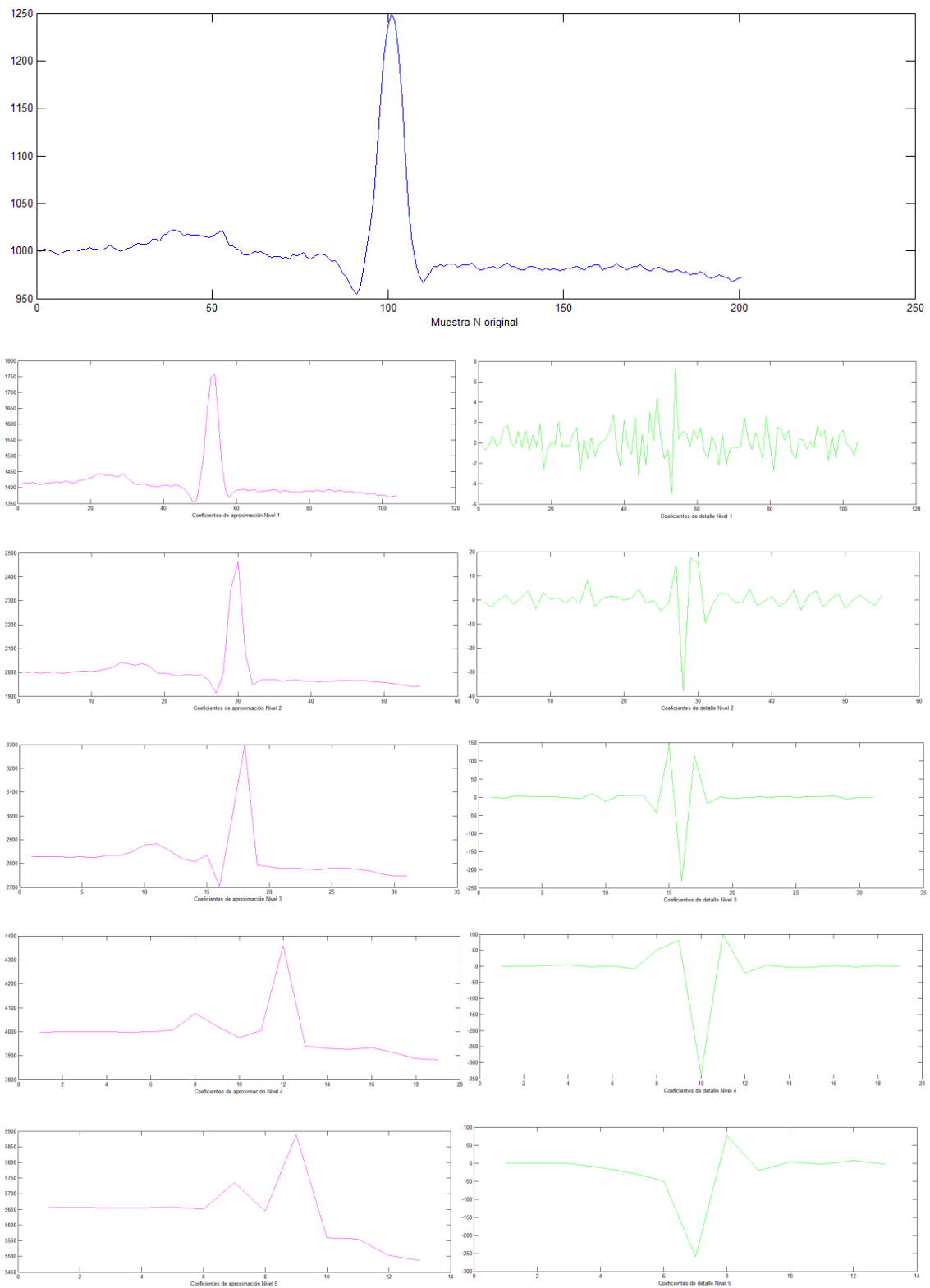


Fig.A.2 Muestra N y sus cinco niveles de descomposición en los coeficientes de aproximación (a la derecha) y los coeficientes de detalle (a la izquierda) mediante base Daubechies 4 de Wavelet.

Al observar la Fig.A.2 se puede deducir que el nivel 4 todavía conserva los detalles principales de la pulsación y a la vez está limpio de ruidos. Como se pretende encontrar un

patrón bastante genérico para cada tipo de pulsación y no detalles concretos, las características de la pulsación serán los coeficientes de aproximación (según referencia [21]).

De igual manera, se ha repetido el mismo procedimiento para la base Daubechies 6. En la Fig. A.3 se representa la descomposición de la señal N en cinco niveles para Db6.

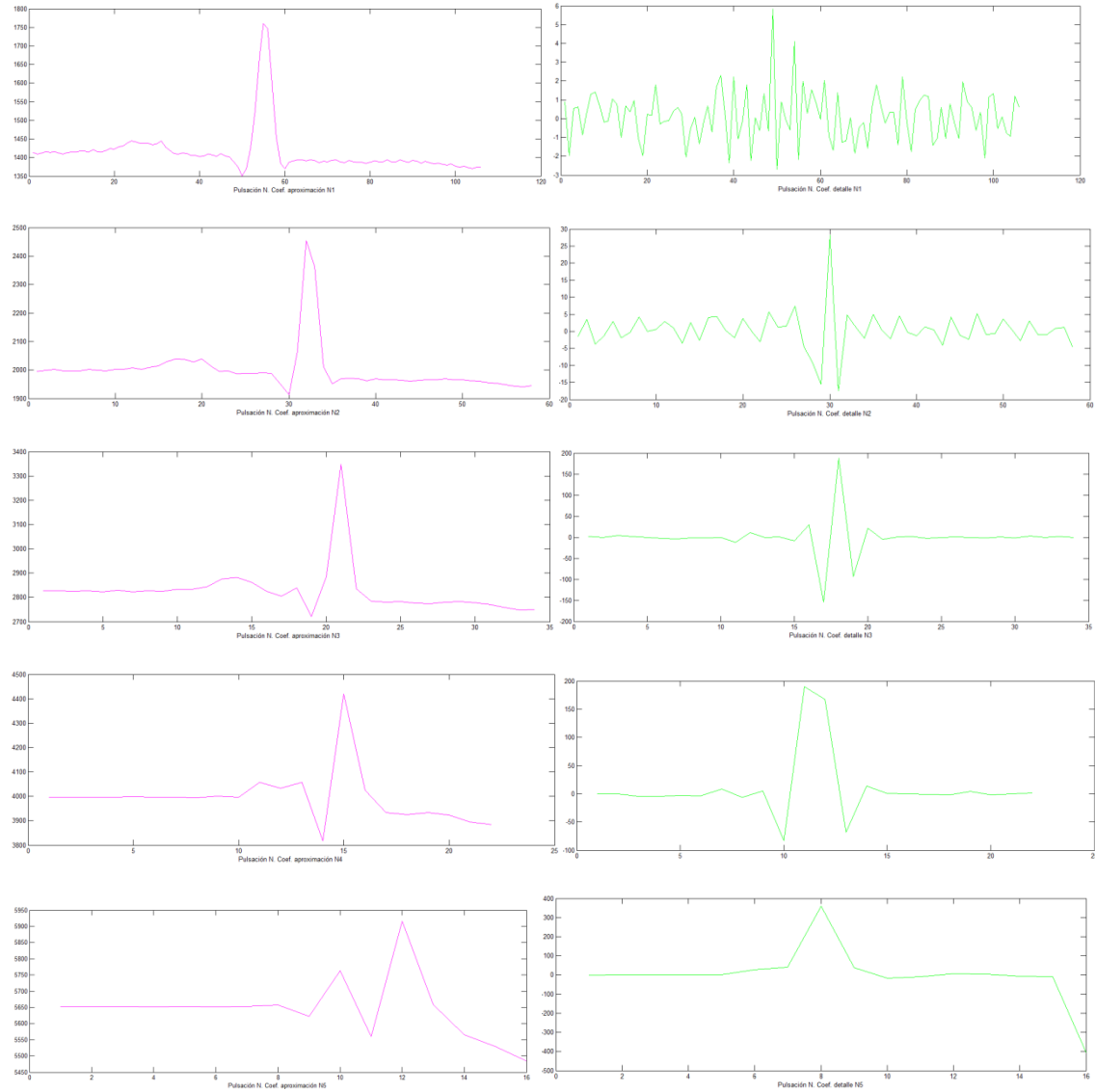


Fig. A.3 . Descomposición de la pulsación N en cinco niveles de coeficientes.